Chapter 9

SIMULATION OF PROBABILITY MODELS

Computational methods for optimization are important, because most optimization problems are too difficult to be solved analytically. For dynamic models, it is often possible to determine steady–state behavior analytically, but the study of transient (time–dependent) behavior requires computer simulation. Probability models are even more complex. Models with no time dynamics can sometimes be solved analytically, and steady–state results are available for the simplest stochastic models. But for the most part, probability models are solved by simulation. In this chapter we will discuss some of the most generally useful simulation methods for probability models.

9.1 Monte Carlo Simulation

Questions involving the transient or time–dependent behavior of stochastic models are difficult to resolve analytically. Monte Carlo simulation is a general modeling technique that is usually effective for such problems. The construction of Monte Carlo simulation software can be time–consuming, and the repeated simulation runs needed for accuracy and sensitivity analysis can become prohibitively expensive. Even so, Monte Carlo simulation models continue to enjoy a very wide appeal. They are easy to conceptualize, easy to explain, and they are the only viable method for the modeling of many complex stochastic systems. A Monte Carlo simulation models random behavior. It can be based on any simple randomizing device, such as a coin flip or the roll of dice, but typically a computer pseudorandom number generator is used. Because of the random element, each repetition of the model will produce different results.

Example 9.1. Arriving on your vacation, you are dismayed to learn that the local weather service forecasts a 50% chance of rain every day this week. What are the chances of 3 consecutive rainy days?

Variables: $X_t = \begin{cases} 0 & \text{if no rain on day } t \\ 1 & \text{if rain on day } t \end{cases}$

Assumptions: $X_1, X_2, \ldots, X_7$ are independent
$\Pr\{X_t = 0\} = \Pr\{X_t = 1\} = 1/2$

Objective: Determine the probability that
$X_t = X_{t+1} = X_{t+2} = 1$ for some
$t = 1, 2, 3, 4,$ or 5

Figure 9.1: Results of step 1 of the rainy day problem.

We will use the five-step method. Step 1 is to formulate a question. In the process we assign variable names to quantities of interest, and we clarify our assumptions about these variables. Then we state a question in terms of these variables. See Figure 9.1 for the results of step 1. Step 2 is to select the modeling approach. We will use Monte Carlo simulation.

Monte Carlo simulation is a technique that can be applied to any probability model. A probability model includes a number of random variables and must also specify the probability distribution for each of these random variables. Monte Carlo simulation uses a randomizing device to assign a value to each random variable, in accordance with its probability distribution. Since the results of the simulation depend on random factors, subsequent repetitions of the same simulation will reproduce different results. Usually a Monte Carlo simulation will be repeated a number of times in order to determine an average or expected outcome.

Monte Carlo simulation is typically used to estimate one or more measures of performance (MOPs) of the system. Repeated simulations can be considered as independent random trials. For the moment let us consider the situation where there is only one simulation parameter Y to be examined. Repeated simulation produces the results $Y_1, Y_2, \ldots, Y_n$, which we may consider as independent and identically distributed random variables whose distribution is unknown. By the strong law of large numbers we know that

$$\frac{Y_1 + \cdots + Y_n}{n} \to EY \tag{9.1}$$

as $n \to \infty$. Hence, we should use the average of $Y_1, \ldots, Y_n$ to estimate the true expected value of Y. We also know that, letting

$$S_n = Y_1 + \cdots + Y_n,$$

the central limit theorem implies that

$$\frac{S_n - n\mu}{\sigma\sqrt{n}}$$

is approximately standard normal for large n, where $\mu = EY$ and $\sigma^2 = VY$. For most cases the normal approximation is fairly good when $n \geq 10$. Even though we do not know μ or σ, the central limit theorem still gives some important insight. The difference between the observed average S_n/n and the true mean $\mu = EY$ is

$$\frac{S_n}{n} - \mu = \frac{\sigma}{\sqrt{n}} \left(\frac{S_n - n\mu}{\sigma\sqrt{n}} \right), \tag{9.2}$$

so we can expect the variation in the observed mean to tend to zero about as fast as $1/\sqrt{n}$. In other words, to get one more decimal place of accuracy in EY would require 100 times as many repetitions of the simulation. More sophisticated statistical analysis is possible, but the basic idea is now very clear. We will have to be satisfied with fairly rough approximations of average behavior if we are to use Monte Carlo simulation.

As a practical matter, there are many sources of error and variation in a modeling problem, and the additional variation produced by Monte Carlo simulation is not typically the most serious of these. A judicious application of sensitivity analysis will suffice to ensure that the results of a simulation are used properly.

Moving on to step 3, we now need to formulate the model. Figure 9.2 gives an algorithm for Monte Carlo simulation of our vacation problem.

As in Chapter 3, the notation Random $\{S\}$ denotes a point selected at random from the set S. In our simulation, each day's weather is represented by a random number from the interval [0, 1]. If the number turns out to be less than p, we assume that this is a rainy day. Otherwise, it is a sunny day. Then p is the probability that any one day is rainy. The variable C simply counts the number of consecutive rainy days. Figure 9.3 shows a slightly modified algorithm. The modified version repeats the Monte Carlo simulation n times and counts the number of rainy weeks (i.e., the number of weeks in which it rains 3 days in a row). The notation

$$Y \leftarrow \text{Rainy Day Simulation } (p)$$

indicates that we evaluate the output variable Y by running the rainy day simulation from Fig. 9.2 with input variable p.

Step 4 is to solve the problem. We ran a computer implementation of the algorithm in Fig. 9.3 with $p = 0.5$ and $n = 100$. The simulation counted 43 rainy weeks out of 100. On this basis we would estimate a 43% chance of a rainy week. Several additional runs were made to confirm these results. In every case the simulation counted around 40 rainy weeks out of 100. Given the likely magnitude of error in the 50% estimated chance of rain, this is about as much accuracy as we will need for this problem. More details on the sensitivity of our simulation results to random factors will be discussed later, in the section on sensitivity analysis.

Algorithm: RAINY DAY SIMULATION

Variables: p = probability of one rainy day

$$X(t) = \begin{cases} 1 & \text{if rain on day } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y = \begin{cases} 1 & \text{if } \geq 3 \text{ consecutive rainy days} \\ 0 & \text{otherwise} \end{cases}$$

Input: p

Process:

```
Begin
Y ← 0
C ← 0
for t = 1 to 7 do
    Begin
    if Random {[0, 1]} < p then
        X(t) = 1
    else
        X(t) = 0
    if X(t) = 1 then
        C ← C + 1
    else
        C ← 0
    if C ≥ 3 then Y ← 1
    End
End
```

Output: Y

Figure 9.2: Pseudocode for Monte Carlo simulation of the rainy day problem.

Finally, step 5. Arriving on your vacation, you find that the local weather service predicts a 50% chance of rain every day for a week. A simulation indicates that, if this forecast is correct, there is a 40% chance that there will be at least 3 consecutive rainy days this week. These results apply to sunshine as well as rain, and so, to end on a somewhat more optimistic note, let us point out that there is a 50% chance of sunshine every day this week, and a 40% chance of at least 3 consecutive days of sunshine. Enjoy your vacation!

We will begin our sensitivity analysis by examining the sensitivity of our simulation results to random factors. Each model run simulates $n = 100$ weeks of vacation and counts the number of rainy weeks. In the terminology of step 2, our MOP is Y, where $Y = 1$ indicates a rainy week, and $Y = 0$ indicates otherwise. Our model simulates $n = 100$ independent random variables $Y_1, \ldots, Y_n$,

Algorithm: REPEATED RAINY DAY SIMULATION

Variables: p = probability of 1 rainy day
n = number of weeks to simulate
S = number of rainy weeks

Input: p, n

Process:
```
Begin
S ← 0
for k = 1 to n do
    Begin
    Y ← Rainy Day Simulation (p)
    S ← S + Y
    End
```

Output: S

Figure 9.3: Pseudocode for repeated Monte Carlo simulation to determine average behavior in the rainy day problem.

all of which have the same distribution as Y. Here $Y_k = 1$ indicates that week k was a rainy week. Our model outputs the random variable $S_n = Y_1 + \cdots + Y_n$, which represents the number of rainy weeks. Let

$$q = \Pr\{Y = 1\} \tag{9.3}$$

denote the probability of a rainy week. It is not hard to calculate that

$$\begin{aligned} \mu &= EY = q \\ \sigma^2 &= VY = q(1-q). \end{aligned} \tag{9.4}$$

Our first model run output $S_n = 43$. On this basis, we can use the strong law of large numbers, Eq. (9.1), to estimate that

$$q = EY \approx S_n/n = 0.43. \tag{9.5}$$

How good is this estimate? By the central limit theorem, we obtain from Eq. (9.2) that S_n/n is unlikely to differ from $\mu = q$ by more than $2\sigma/\sqrt{n}$, since a standard normal random variable is 95% certain to have absolute value less than 2. Using Eqs. (9.4) and (9.5), we would conclude that our estimate in Eq. (9.5) is within

$$2\sqrt{(0.43)(0.57)/100} \approx 0.1 \tag{9.6}$$

of the true value of q.

A more elementary way to investigate the sensitivity of our simulation results to random factors is to compare the results of a number of model runs. Figure

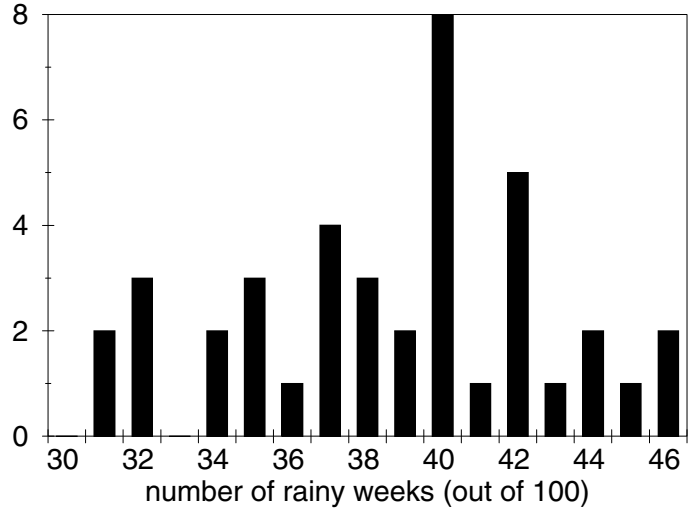


Figure 9.4: Histogram showing the distribution of the number of rainy weeks out of 100 for the rainy day problem.

9.4 shows the results of 40 model runs, each of which simulates 100 weeks of vacation. All of these model runs lead to estimates $S_n/n \approx 0.4$, and none is outside of the interval 0.4 ± 0.1. It also appears that the distribution of S_n is approximately normal.

We should also examine the sensitivity of our simulation results to the forecast 50% chance of rain. Figure 9.5 shows the results of 10 additional model runs for each of the cases $p = 0.3$, 0.4, 0.5, 0.6, and 0.7, where p is the probability of one rainy day. The boxes connected by the dotted line show the average outcome (fraction of rainy weeks), and the vertical bars show the range of outcomes for each case. For a 40% chance of rain each day, the probability of at least 3 consecutive rainy days in a week is around 20%, and so forth. While the probability of 3 straight days of rain varies quite a bit, it seems safe to say that if the chance of rain each day is moderate, then so is the chance of rain 3 straight days in one week.

What about robustness? We should examine the critical assumptions that make up the structure of the model. In step 1 we assumed that the indicator variables $X_1, \ldots, X_7$ were independent and identically distributed. In other words, the chance of rain is the same each day, and the weather on one day is independent of the weather on any other day. Suppose instead that $\Pr\{X_t = 1\}$ varies with t, still keeping the independence assumption. We can use our sensitivity analysis results to obtain upper and lower bounds on the probability of 3 consecutive days of rain, reasoning that this probability should be monotone

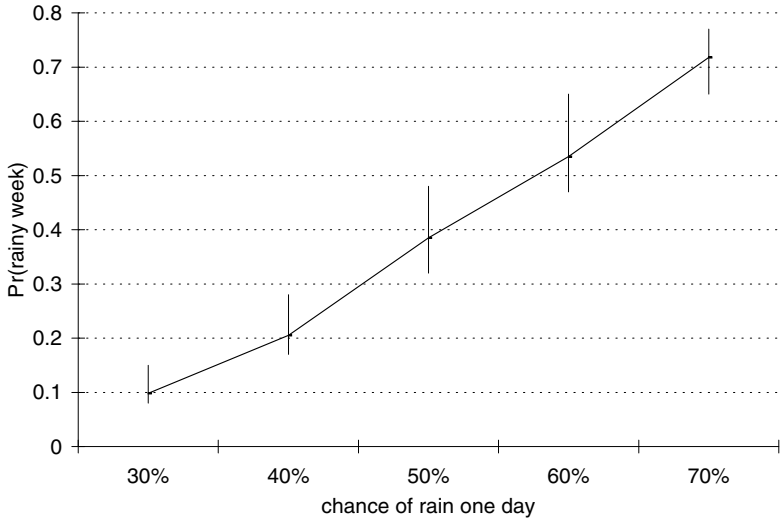


Figure 9.5: Graph of the probability of a rainy week versus the chance of rain one day for the rainy day problem.

increasing as any $\Pr\{X_t = 1\}$ increases. A higher chance of rain on day t means a higher probability of 3 straight days of rain, all else being equal. Then, if all $\Pr\{X_t = 1\}$ are between 0.4 and 0.6 (a 40% to 60% chance of rain), the probability of 3 days of rain is between 0.2 and 0.6. Our model is quite robust in this regard, partly due to the fact that we are not looking for a lot of accuracy in our answer.

Suppose now that $\{X_t\}$ are not independent. For example, we could model $\{X_t\}$ as a Markov chain on the state space $\{0, 1\}$. This implies that the chance of rain today depends on the weather yesterday. Local weather forecasts rarely contain the kind of information necessary to formulate such a model, particularly the state transition probabilities. However, we could always guess at these and then use sensitivity analysis to ensure that we have not assumed too much. The kind of question we are asking here cannot be answered using a steady–state analysis, because it concerns the time–dependent or transient behavior of the stochastic process. Such questions are usually difficult to handle analytically, even by advanced techniques. This is one reason why Monte Carlo simulation is so widely used. On the other hand, it is sometimes possible to develop analytical solutions; see Exercise 9.19.

9.2 The Markov Property

A stochastic process is said to have the *Markov property* if the information contained in the current state of the process is all that is needed to determine the probability distribution of future states. Markov chains and Markov processes were introduced in Chapter 8. Both have the Markov property. Monte Carlo simulation of a stochastic process is much simpler in the presence of the Markov property, because it reduces the quantity of information that needs to be stored in the computer.

Example 9.2. We reconsider the docking problem of Example 4.3, now taking into account the random element. Our basic assumptions are summarized in Fig. 4.7. Our goal or objective, as before, is to determine the success of our control procedure to match velocities.

We will use the five-step method. Our starting point is Fig. 4.7, but we must also make some assumptions about the variables a_n, c_n, and w_n. Ideally we would perform some experiments and collect data on such factors as astronaut response time and the time it takes to read or manipulate controls. In the absence of such data, we would attempt to make some reasonable assumptions consistent with what is known about similar situations.

The random variable that represents the most uncertainty (i.e., the greatest variance) is c_n, the time it takes to make a control adjustment. This variable represents the time it takes to observe the rate of closing, calculate the desired acceleration adjustment, and carry out the adjustment. We will assume that it takes roughly 1 second to observe the rate of closing, 2 seconds to calculate the adjustment, and 2 seconds to make the adjustment. The actual time to carry out each phase is random. Let R_n denote the time to read the closing velocity, S_n the time to calculate the desired adjustment, and T_n the time to make the adjustment. Then $ER_n = 1$ second and $ES_n = ET_n = 2$ seconds. We need to make a reasonable assumption about the distributions of these random variables. It seems reasonable to suppose that they are all nonnegative, mutually independent, and that outcomes close to the mean are most likely. There is a wide variety (infinite!) of distributions that fit this general description, and as of now, we have no particular reason to prefer one over another. This being the case, we will refrain from specifying the exact distribution at this time. One of our random variables is $c_n = R_n + S_n + T_n$. The others are w_n, the waiting time before the next control adjustment, and a_n, the acceleration after this control adjustment. We will assume that $a_n = -kv_n + \varepsilon_n$, where ε_n is a (small) random error. We assume that ε_n is normally distributed with mean zero. The variance of ε_n depends on both the skill of the human operator and on the sensitivity of the control mechanism. We will assume that we can typically achieve an accuracy of about $\pm$ 0.05 m/sec^2, and so we set the standard deviation of ε_n at $\sigma = 0.05$. The waiting time w_n will depend on c_n if we are trying to maintain a fixed time between control adjustments of 15 seconds total. We will assume that $w_n = 15 - c_n + E_n$, where E_n is a small random error. Assume E_n

Variables:
t_n = time of nth velocity observation (sec)
v_n = velocity at time t_n (m/sec)
c_n = time to make nth control adjustment (sec)
a_n = acceleration after nth control adjustment (m/sec^2)
w_n = wait before $(n+1)$st observation (sec)
R_n = time to read velocity (sec)
S_n = time to calculate adjustment (sec)
T_n = time to make adjustment (sec)
ε_n = random error in control adjustment (m/sec^2)
E_n = random error in waiting time (sec)

Assumptions:
$t_{n+1} = t_n + c_n + w_n$
$v_{n+1} = v_n + a_{n-1}c_n + a_n w_n$
$a_n = -kv_n + \varepsilon_n$
$c_n = R_n + S_n + T_n$
$w_n = 15 - c_n + E_n$
$v_0 = 50,\ t_0 = 0$
$ER_n = 1,\ ES_n = ET_n = 2$, and the distribution of $R_n,\ S_n,\ T_n$ is yet to be specified
ε_n is normal mean 0, standard deviation 0.05
E_n is normal mean 0, standard deviation 0.1

Objective:
Determine $T = \min\{t_n : |v_n| \leq 0.1\}$

Figure 9.6: Results of step 1 of the docking problem with random factors.

has mean zero, is normal, and that astronaut response–time limitations imply a standard deviation of 0.1 seconds.

We should now consider the analysis objectives. We want to determine the success of our control procedure. Certainly, we are interested in seeing that $v_n \to 0$. Simulation can determine that. But we also have the opportunity to gather information on other aspects of system performance. We should decide at this time, as a part of step 1, what measures of performance (MOPs) we want to track. The selected MOPs should represent important quantitative information that can be used as the basis for comparison with computing control procedure options. Assume that our initial closing velocity is 50 m/sec, and that the velocity–matching process is considered successful when the closing velocity has been reduced to 0.1 m/sec. We would be most interested in the total time it takes to succeed. This will be our measure of performance. At this point, to conclude step 1, we summarize our results in Figure 9.6.

Step 2 is to select the modeling approach. We will use a Monte Carlo simulation based on the Markov property.

The general idea is as follows. At each time step n there is

```
Begin
Read data
Initialize X_0
While (not done) do
  Begin
  Determine distribution of X_{n+1} using X_n
  Use Monte Carlo method to determine X_{n+1}
  Update records for MOPs
  End
Calculate and output MOPs
End
```

Figure 9.7: Algorithm for the general Markovian simulation.

a vector X_n that describes the current state of the system. The sequence of random vectors $\{X_n\}$ is assumed to have the Markov property. In other words, the current state X_n contains all of the information needed to determine the probability distribution of the next state, X_{n+1}.

The general structure of the simulation is as follows. First, we initialize variables and read data files. At this stage we must specify the initial state X_0. Next, we enter a loop that repeats until an end condition is satisfied. In the loop we use X_n to specify the distribution of X_{n+1}, and then we use a random number generator to determine X_{n+1} according to that distribution. We must also calculate and store any information needed to generate the simulation MOPs. Once the end condition occurs, we exit the loop and output the MOPs. Then we are done. The algorithm for this simulation is illustrated by Figure 9.7.

We need to discuss the inner loop in some detail. Suppose for now that the state vector X_n is one–dimensional. Let

$$F_\Theta(t) = \Pr\{X_{n+1} \leq t | X_n = \Theta\}.$$

The value of $\Theta = X_n$ determines the probability distribution of X_{n+1}. The function F_Θ maps the state space

$$E \subseteq \mathbb{R}$$

onto the interval [0, 1]. There are widely available methods for generating a random number in [0, 1], and these can be used to generate a random variable with distribution F_Θ. Since

$$y = F_\Theta(x)$$

maps

$$E \to [0, 1],$$

the inverse function

$$x = F_\Theta^{-1}(y)$$

maps

$$[0, 1] \to E.$$

If U is a random variable uniformly distributed over $[0, 1]$ (i.e., the density function of U is the function equal to 1 on $[0, 1]$ and equal to 0 elsewhere), then $X_{n+1} = F_\Theta^{-1}(U)$ has distribution F_Θ, since given $X_n = \Theta$,

$$\begin{aligned} \Pr\{X_{n+1} \le t\} &= \Pr\{F_\Theta^{-1}(U) \le t\} \\ &= \Pr\{U \le F_\Theta(t)\} \\ &= F_\Theta(t) \end{aligned} \tag{9.7}$$

because

$$\Pr\{U \le x\} = x \text{ for } 0 \le x \le 1.$$

Example 9.3. Let $\{X_n\}$ denote a stochastic process where X_{n+1} has an exponential distribution with rate parameter X_n. Determine the first passage time

$$T = \min\{n : X_1 + \cdots + X_n \ge 100\},$$

assuming that $X_0 = 1$.

We will present a computer simulation to solve this problem once we discuss the details of generating X_{n+1} from X_n. Letting $\Theta = X_n$, the density function of X_{n+1} is

$$f_\Theta(x) = \Theta e^{-\Theta x}$$

on $x \ge 0$. The distribution function is

$$F_\Theta(x) = 1 - e^{-\Theta x}.$$

Setting

$$y = F_\Theta(x) = 1 - e^{-\Theta x}$$

and inverting, we obtain

$$x = F_\Theta^{-1}(y) = -\ln(1 - y)/\Theta.$$

Hence we can let

$$X_{n+1} = -\ln(1 - U)/\Theta,$$

where U is a random number between 0 and 1. See Figure 9.8 for the complete simulation algorithm.

The discussion above provides a method of generating random variables with any prescribed distribution. While useful in theory,

Algorithm: FIRST PASSAGE TIME SIMULATION (Example 9.3)

Variables: X = initial state variable
N = first passage time

Input: X

Process:
```
Begin
S ← 0
N ← 0
until (S ≥ 100) do
    Begin
    U ← Random {[0, 1]}
    R ← X
    X ← − ln(1 − U)/R
    S ← S + X
    N ← N + 1
    End
End
```

Output: N

Figure 9.8: Pseudocode for the Markovian simulation for Example 9.3.

in practice there is sometimes a catch. For many distributions, such as the normal distribution, it is not easy to compute the inverse function F_Θ^{-1}. We can always circumvent this difficulty by interpolating from a table of functional values, but in the case of a normal distribution, there is an easier way.

The central limit theorem guarantees that for any sequence of independent, identically distributed random variables $\{X_n\}$ with mean μ and variance σ^2, the normalized partial sums

$$\frac{(X_1 + \cdots + X_n) - n\mu}{\sigma\sqrt{n}}$$

tend to a standard normal distribution. Suppose $\{X_n\}$ are uniform on [0, 1]. Then

$$\begin{aligned} \mu &= \int_0^1 x \cdot dx = 1/2 \\ \sigma^2 &= \int_0^1 (x - 1/2)^2 \, dx = 1/12, \end{aligned} \tag{9.8}$$

and so for n sufficiently large, the random variable

$$Z = \frac{(x_1 + \cdots + X_n) - n/2}{\sqrt{n/12}} \tag{9.9}$$

Algorithm: NORMAL RANDOM VARIABLE

Variables: μ = mean
σ = standard deviation
Y = normal random variable with mean μ, standard deviation σ

Input: μ, σ

Process:
```
Begin
S ← 0
for n = 1 to 12 do
    Begin
    S ← S + Random {[0, 1]}
    End
Z ← S − 6
Y ← μ + σZ
End
```

Output: Y

Figure 9.9: Pseudocode for Monte Carlo simulation of a normal random variable.

is approximately standard normal. For most purposes a value of $n \geq 10$ is sufficient. We will use $n = 12$ to eliminate the denominator in Eq. (9.9). Given a standard normal random variable Z, another normal random variable Y with mean μ and standard deviation σ can be obtained by setting

$$Y = \mu + \sigma Z. \tag{9.10}$$

Figure 9.9 shows a simple algorithm for generating normal random variables with a specified mean and variance.

Returning to the docking problem, we begin step 3. Our first concern is to identify our state variables. In this case we can take

$$\begin{aligned} T &= t_n \\ V &= v_n \\ A &= a_n \\ B &= a_{n-1} \end{aligned} \tag{9.11}$$

as our state variables. Since our only MOP is already a state variable, we will not need to initialize or update any additional variables for that purpose. Figure 9.10 gives the algorithm for our docking simulation. The notation Normal (μ, σ) denotes the output of the normal random variable algorithm described in Fig. 9.9.

Algorithm: Docking Simulation

Variables:
k = control parameter
n = number of control adjustments
$T(n)$ = time (sec)
$V(n)$ = current velocity (m/sec)
$A(n)$ = current acceleration (m/sec^2)
$B(n)$ = previous acceleration (m/sec^2)

Input: $T(0),\ V(0),\ A(0),\ B(0),\ k$

Process:
Begin
$n \leftarrow 0$
while $|V(n)| > 0.1$ do
    Begin
    $c \leftarrow$ Normal (5, 1)
    $B(n) \leftarrow A(n)$
    $A(n) \leftarrow$ Normal $(-k\,V(n),\ 0.05)$
    $w \leftarrow$ Normal $(15 - c,\ 0.1)$
    $T(n) \leftarrow T(n) + c + w$
    $V(n) \leftarrow V(n) + c\,B(n) + w\,A(n)$
    $n \leftarrow n + 1$
    End
End

Output: $T(n)$

Figure 9.10: Pseudocode for Monte Carlo simulation of the docking problem.

For now we will assume that

$$c_n = R_n + S_n + T_n$$

has a normal distribution with a mean of $\mu = 5$ seconds and a standard deviation of $\sigma = 1$ second.

Figure 9.11 shows the results of 20 simulation runs. In these runs the docking time ranged between 156 and 604 seconds, with an average docking time of 305 seconds.

We have constructed a Monte Carlo simulation of a velocity–matching exercise for spacecraft manual docking. Our simulation takes into account the random factors inherent in this man–machine system. Based on what we believe to be a reasonable set of assumptions on operator performance, the model indicates a wide variance in the time to complete the docking procedure. For example, the time to match velocities starting from a relative velocity of 50 m/sec, and using a 1 : 50 control factor, averaged about 5 minutes. But outcomes of less than 3 or greater than 7 minutes are not uncommon. The major source

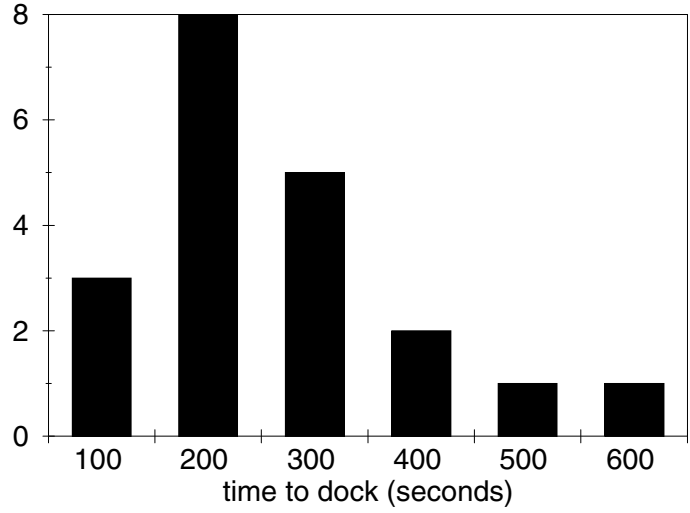


Figure 9.11: Histogram showing the distribution of docking times for the docking problem: case $k = 0.02$, $\sigma = 0.1$.

of variation is the time it takes the pilot to complete the control adjustment procedure.

One important parameter for sensitivity analysis is the standard deviation of c_n, the time to make a control adjustment. We have assumed that the standard deviation of c_n is $\sigma = 1$. Figure 9.12 shows the results of 20 additional model runs for a few values of σ near 1. As in Fig. 9.5, we let the vertical bars represent the range of outcomes, and the boxes connected by the dotted line indicate the average outcome for each value of σ. It seems that our overall conclusions are fairly insensitive to the exact value of σ. In every case the average docking time is around 300 seconds (5 minutes), and the variation in docking times is quite large.

Probably the most important parameter for sensitivity analysis is the control parameter k. We have assumed $k = 0.02$, which results in an average docking time of around 300 minutes. Figure 9.13 shows the results of some additional model runs in which we varied k. We made 20 additional model runs for each new value of k. As before, the vertical bars denote the range of outcomes, and the boxes represent the average outcome for each value of k. As we would expect, docking time is reasonably sensitive to the control parameter k. Of course, it is of considerable interest to determine the best value of k. We will leave this problem for the exercises.

The main robustness question for this model is the distribution of c_n, which we assumed was normal. Since our results did not vary significantly with small

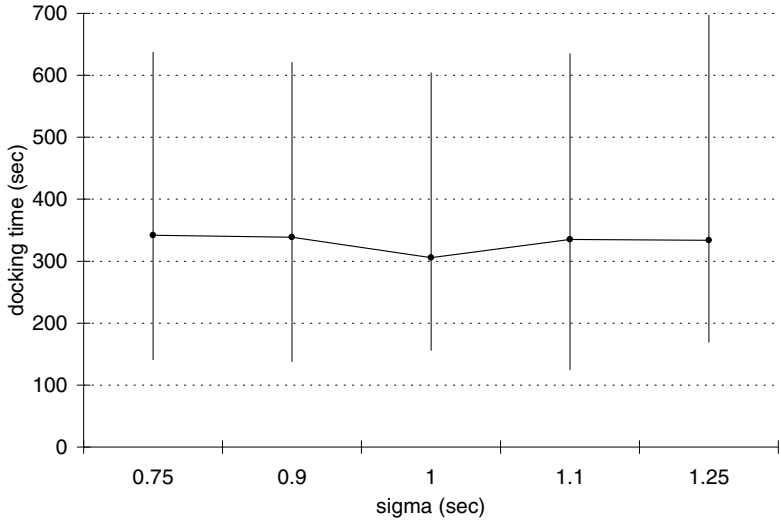


Figure 9.12: Graph of docking time versus standard deviation σ of time to make nth control adjustment for the docking problem.

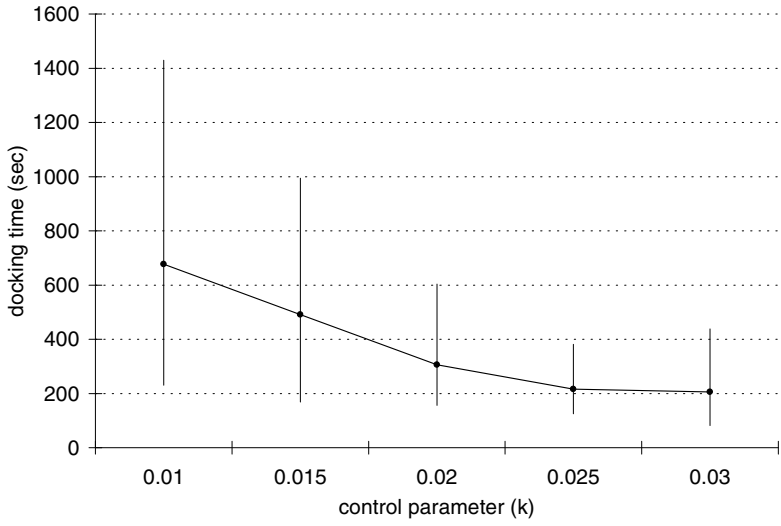


Figure 9.13: Graph of docking time versus control parameter k for the docking problem: case $\sigma = 1$.

changes in σ, and since we are not demanding much accuracy, there is ample reason to expect the model to exhibit robustness with respect to the distribution of c_n. There are a few simple experiments to verify robustness that suggest themselves. We leave these to the exercises.

9.3 Analytic Simulation

Monte Carlo simulation models are relatively easy to formulate, and they are intuitively appealing. Their major drawback is that a very large number of model runs is required to obtain reliable results, especially in the area of sensitivity analysis. Analytic simulations are more difficult to formulate, but are computationally more efficient.

Example 9.4. A military operations analyst plans an air strike against a well–defended target. High–altitude strategic bombers will be sent to attack this important target. It is important to ensure the success of this attack early in the battle, preferably on the first day. Each individual aircraft has a 0.5 probability of destroying the target, assuming that it can get through the air defense and then acquire (i.e., find) the target. The probability that a single aircraft will acquire the target is 0.9. The target is defended by 2 surface–to–air missile (SAM) batteries and a number of air defense guns. The flight profile of the aircraft will prevent the air defense guns from being effective (because the planes will be too high). Each SAM battery has its own tracking radar and computer guidance equipment, which is capable of tracking 2 aircraft and guiding 2 missiles simultaneously. Intelligence estimates a 0.6 probability that 1 missile will disable its target aircraft. Both SAM batteries share a target acquisition radar that is highly effective against high–altitude bombers at a range of up to 50 miles. The effective range of the tracking radar is 15 miles. The bombers will travel at 500 miles/hour at an altitude of 5 miles, and the attack requires that they loiter in the target area for 1 minute. Each SAM battery can launch 1 missile every 30 seconds, and the missiles travel at 1,000 miles/hour. How many bombers should be sent against this target to ensure its destruction?

We will use the five-step method. Step 1 is to formulate a question. We want to know how many aircraft to commit to this mission. The goal is to destroy the target, but it is immediately clear that we cannot demand a 100% guarantee of success. Let us say now that we want to be 99% sure of destroying the target. We will perform a sensitivity analysis on this number later on. Suppose that N aircraft are committed to this mission. The number of planes destroyed by air defenses before they can complete their attack on the target is a random variable. We will denote this random variable by X. In order to obtain an expression for the probability S that the mission is a success, we will proceed in two stages. First we will obtain an expression for the probability P_i of mission success, given that $X = i$ planes are destroyed before they can attack the target.

Then we will obtain the probability distribution of X so that we can compute

$$S = \sum_i P_i \Pr\{X = i\}. \tag{9.12}$$

If N aircraft are sent and $X = i$ are destroyed before they reach the target, then there are $(N - i)$ attacking aircraft. If p is the probability that 1 attacking aircraft can destroy the target, then $(1 - p)$ is the probability that 1 attacking aircraft will fail to destroy the target. The probability that all $(N - i)$ attacking aircraft fail to destroy the target is

$$(1 - p)^{N-i}.$$

Thus, the probability that at least one of the attacking aircraft succeeds in destroying the target is

$$P_i = 1 - (1 - p)^{N-i}. \tag{9.13}$$

The total exposure time to air defense prior to completing the attack is

$$\frac{15 \text{ mi}}{500 \text{ mi/hr}} \cdot \frac{60 \text{ min}}{\text{hr}} = 1.8 \text{ min}$$

on the way to the target, and an additional minute in the target area, for a total of 2.8 minutes, in which time each SAM battery fires 5 shots. Thus, the attacking aircraft will be exposed to $m = 10$ shots total. We assume the number of aircraft X destroyed will have a *binomial distribution*

$$\Pr\{X = i\} = \binom{m}{i} q^i (1 - q)^{m-i}, \tag{9.14}$$

where

$$\binom{m}{i} = \frac{m!}{i!(m - i)!} \tag{9.15}$$

is the binomial coefficient. (The distribution in Eq. (9.14) is the analytic model for the number of successes in m trials, where q represents the probability of success. See Exercise 12 for more details.) Now, in order to compute the probability of mission success S, we need to substitute Eqs. (9.13) and (9.14) back into Eq. (9.12). Then our objective is to determine the smallest N for which $S > 0.99$. This concludes step 1. We summarize our results in Figure 9.14.

Step 2 is to select our modeling approach. We will use an *analytic simulation* model. In a Monte Carlo simulation we draw random numbers to simulate events and use repeated trials to estimate probabilities and expected values. In an analytic simulation we use a combination of probability theory and computer programming to calculate probabilities and expected values. Analytic simulations are more mathematically sophisticated, which accounts for their greater efficiency. The feasibility of analytic simulation depends on both the problem complexity and the skill of the modeler. Most highly skilled analysts consider

Variables:
N = number of bombers sent
m = number of missiles fired
p = probability 1 bomber can destroy target
q = probability 1 missile can disable bomber
X = number of bombers disabled prior to attack
P_i = probability of mission success given $X = i$
S = overall probability of mission success

Assumptions:
$p = (0.9)(0.5)$
$q = 0.6$
$m = 10$
$P_i = 1 - (1 - P)^{N-i}$
$\Pr\{X = i\} = \binom{m}{i} q^i (1-q)^{m-i}; i = 0, 1, 2, \ldots, m$
$S = \sum_{i=0}^{m} P_i \Pr\{X = i\}$

Objective:
Find the smallest N for which $S > 0.99$

Figure 9.14: Results of step 1 of the bombing run problem.

Monte Carlo simulation as a last resort, to be employed only if they are unable to formulate a suitable analytic model.

Step 3 is to formulate the model. Figure 9.15 shows an algorithm for analytic simulation of the bombing run problem. The notation Binomial (m, i, q) denotes the value of the binomial probability defined by Eqs. (9.14) and (9.15).

Step 4 is to solve the model. We used a computer implementation of the algorithm in Fig. 9.15, with inputs

$$\begin{aligned} m &= 10 \\ p &= (0.9)(0.5) \\ q &= 0.6, \end{aligned}$$

and we varied N to obtain the results shown in Figure 9.16.

A minimum of $N = 15$ planes is required to ensure a 99% chance of mission success. Step 5 is to answer the question, which is how many bombers we must commit to this mission in order to be 99% sure of success. The answer is 15. Now we need to conduct a sensitivity analysis to get at the broader question of what would be a good number of bombers to commit to this mission.

First let us consider the desired probability of success $S = 0.99$. This is a number we literally just made up. Figure 9.17 shows the effect of varying this parameter. It appears that $N = 15$ is a reasonable decision, although any number greater than 10 and less than 20 would be fine. Sending more than 20 aircraft would be overkill.

Bad weather would decrease the detection probability, which factored into the equation $p = (0.9)(0.5)$. If the detection probability decreases to 0.5, then $p = 0.25$, and for $S > 0.99$ we will need at least $N = 23$ aircraft. If the detection

Algorithm: BOMBING RUN PROBLEM

Variables: N = number of bombers sent
m = number of missiles fired
p = probability 1 bomber can destroy target
q = probability 1 missile can disable bomber
S = probability of mission success

Input: N, m, p, q

Process:
Begin
$S \leftarrow 0$
for $i = 0$ to m do
    Begin
    $P \leftarrow 1 - (1 - p)^{N-i}$
    $B \leftarrow$ Binomial (m, i, q)
    $S \leftarrow S + P \cdot B$
    End
End

Output: S

Figure 9.15: Pseudocode for analytic simulation of the bombing run problem.

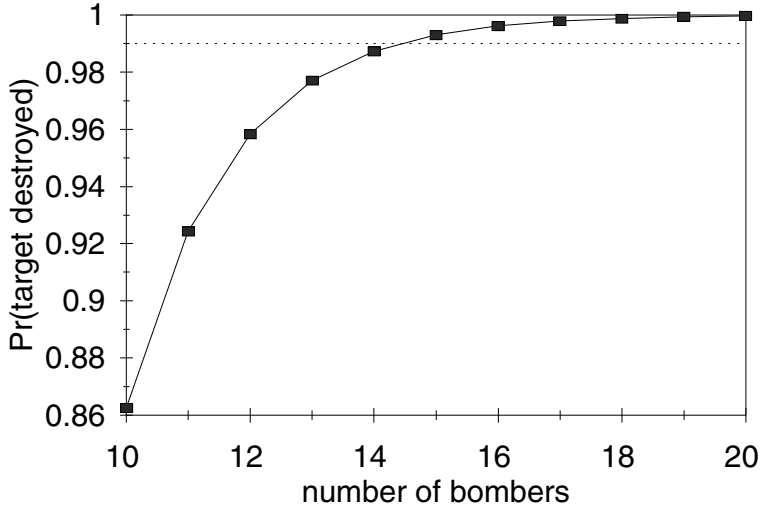


Figure 9.16: Graph of mission success probability S versus number of bombers sent N for the bombing run problem.

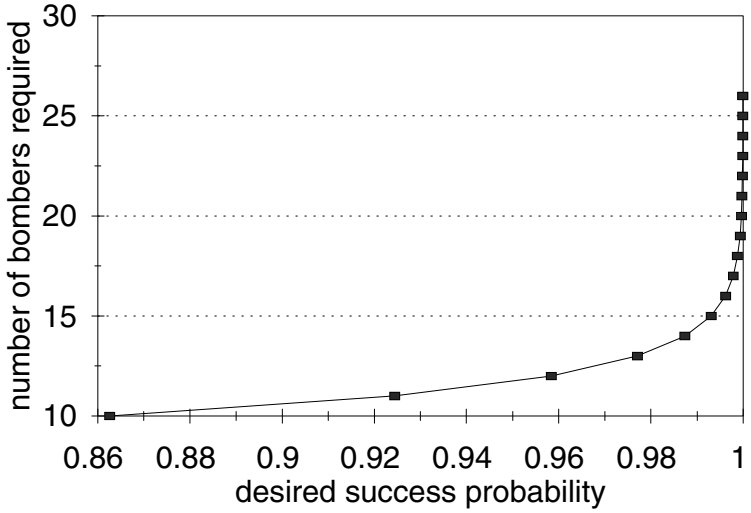


Figure 9.17: Graph of minimum number of bombers N required to obtain mission success probability S in the bombing run problem.

probability is 0.3, then $N = 35$ aircraft are required. The overall relationship between the detection probability and the number of planes required for mission success is illustrated in Figure 9.18. It is unlikely that we will want to fly this mission in bad weather.

One of the applications for models of this kind is to analyze the potential operational impact of engineering advances. Suppose we had a bomber that flew at 1,200 miles/hour and that reduced the loitering time at the target to 15 seconds. Now the aircraft will be exposed to SAM fire for only 1 minute, so that the air defense can only shoot $m = 4$ missiles. Now $N = 11$ bombers are required to get a 99% chance of success. Figure 9.19 shows the relationship between the number of bombers sent and the probability of mission success in the baseline case ($m = 10$ missiles are fired by the air defense) and in the case of the advanced concept aircraft ($m = 4$). For the sake of comparison, we also include the case $m = 0$ missiles fired. This curve represents the maximum potential benefit of the proposed technology. If bombers are exposed to no threat from air defense, it still takes at least 8 planes to ensure a 99% chance of success.

Suppose that we would produce a better targeting system that increased the probability of one aircraft destroying the target to 0.8. Now $p = (0.9)(0.8) = 0.72$ and, all else remaining the same, it would take $N = 13$ aircraft to achieve $S > 0.99$. This is not much of an improvement. If we combine the high–speed bomber with the high–accuracy bombs, we can trim the number of planes needed to 8.

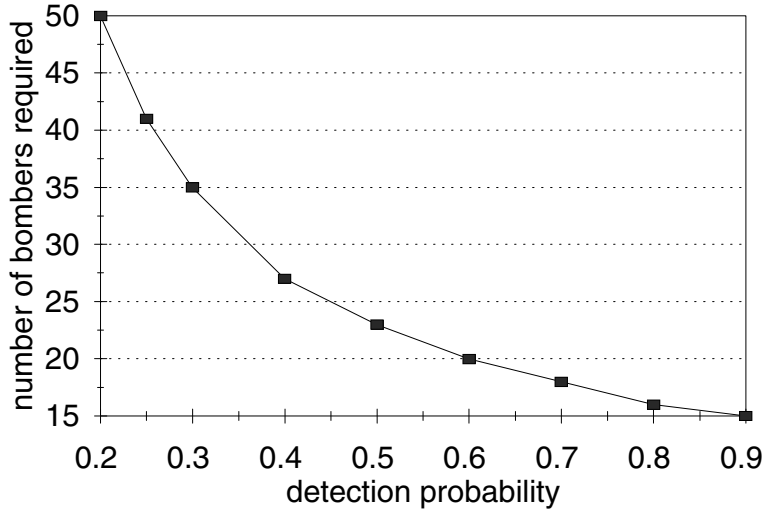


Figure 9.18: Graph of minimum number of bombers N required for mission success versus probability bombers can detect target for the bombing run problem.

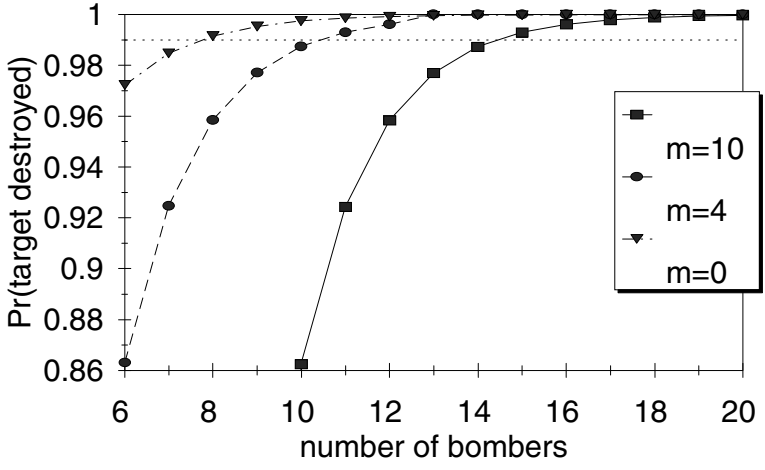


Figure 9.19: Graph of probability of mission success S versus number of bombers sent N for the bombing run problem: comparison of the three cases where the number of missiles fired is $m = 0$, 4, and 10.

What if we have underestimated the effectiveness of enemy air defenses? If $q = 0.8,$ then $N = 17$ aircraft are required for $S > 0.99$. If $q = 0.6$ but $m = 15$ (assume that there are 3 SAM batteries), then 18 planes are needed. In either case our general conclusions are not seriously altered.

On the subject of robustness, we have made a number of simplifying assumptions in our model. We have assumed that each of the attacking aircraft acquires the target independently and with the same probability. In reality the first bombs may throw up smoke and dust, obscuring the target area. This may reduce the probability that the remaining bombers can acquire the target. We have already performed a sensitivity analysis on this parameter and found that N is quite sensitive to this probability. Although our model cannot represent a dependence here, it does provide bounds. If each individual aircraft has at least a 50% chance of target acquisition, then $N = 23$ planes will suffice.

Our model also assumed that the SAM batteries never shoot twice at the same aircraft. This is certainly the optimal strategy in a target–rich environment, where the number of potential targets exceeds the maximum number of shots. But suppose that a new stealth technology is able to reduce significantly the number of aircraft detected. Now the air defense may have more shots than targets. Assume that the air defense is able to tell whether they have disabled an aircraft, so that they will not waste any shots. If d aircraft are detected, then the number of aircraft kills in m shots may be represented using a Markov chain model where

$$X_n \in \{0, 1, \ldots, d\}$$

is the number of hits after n shots. The state transition probabilities are

$$\Pr\{X_{n+1} = i + 1 | X_n = i\} = q \text{ and } P_r\{X_{n+1} = i | X_n = i\} = 1 - q$$

for $0 \leq i < d$, and of course

$$\Pr\{X_{n+1} = d | X_n = d\} = 1.$$

If $X_0 = 0,$ then X_m is the number of aircraft lost. It would be possible to calculate the probability distribution of X_m for each $d = 1, \ldots, N$ and incorporate this into a modified version of our model. This is an example of the application of transient analysis for a Markov chain. It is also much easier to implement a Monte Carlo simulation for this problem. See Exercises 14 and 15 below. Our analytic simulation model is not very robust in this regard.

9.4 Particle Tracking

Particle tracking is a method for solving partial differential equations by simulating the associated stochastic process. This Monte Carlo simulation method is much simpler to code than other numerical solution methods. It is particularly useful for models with variable coefficients, irregular domains, or boundary values. For diffusion problems, it also provides a useful physical model for the motion of individual particles.

Variables:	t = Time since release of pollutant (hrs) d = Distance between pollution particle and town (km) s = Plume spread at time t (km) P = Pollution concentration in town (ppm) v = Wind speed at distance d from town (km/hr)
Assumptions:	$v = 3$ if $d > 10$ $v = 8 - 5d/10$ if $d \leq 10$ Peak concentration $P = 1000$ ppm at $t = 1$ hour Plume spread is $s = 2000$ meters at $t = 1$ hour
Objective:	Determine the maximum pollution level in town, and time until pollution falls to safe level of 50 ppm.

Figure 9.20: Results of Step 1 of the pollution problem.

Example 9.5. Reconsider the pollution problem of Example 7.5, but now consider that the wind speed increases as we get closer to the city. Suppose that the wind speed increases from 3 kilometers per hour at the spill site to 8 kilometers per hour in the center of town. This is due to a *heat island* effect, where the buildings and pavement in the city retain more heat, leading to warmer conditions and higher wind speeds. Taking this into account, what is the maximum concentration expected in town, when will it occur, and how long until the concentration of pollutant falls back below a safe level? For this Category 1 pollutant (the most dangerous kind), the US Environmental Protection Agency safe level is 50 parts per million (ppm) by volume.

We will use the five-step method. The first step is to ask a question. We want to know the concentration of pollutant in town and how it varies over time. We will assume that the wind speed varies depending on the distance from the center of town. Since we do not know how the actual wind speed varies, we will make the simple assumption that wind speed varies linearly between a value of 3 km/hr at distances 10 or more km from town, up to 8 km/hr at the center of town. Then later we will test this assumption through sensitivity and robustness analysis (see also Exercise 21 at the end of this chapter). The results of Step 1 are summarized in Figure 9.20.

Step 2 is to select the modeling approach. We will use a diffusion model, and solve the model by the method of particle tracking.

The diffusion equation

$$\frac{\partial C}{\partial t} = \frac{D}{2}\frac{\partial^2 C}{\partial x^2}, \tag{9.16}$$

was introduced in Section 7.4. Now we modify this equation to explicitly represent plume velocity. Recall that $C(x,t)$ represents the relative concentration of contaminants at location x at time t.

The diffusion equation (9.16) models spreading away from the plume center of mass, and for a constant velocity, the center of mass is located at $\mu = vt$ where v is the average velocity. A probability model for the diffusion equation was also explained in Section 7.4. In that model, we track the particle plume in a moving coordinate system, with the origin at the plume center of mass. Each particle makes a small random movement X_i over a small time interval Δt, and these movements are assumed to be independent with mean 0 and variance $D\Delta t$. Then the central limit theorem implies that the particle has deviated by an amount $X_1+\cdots+X_n \approx \sqrt{Dt}Z$ away from the center of mass by time $t = n\Delta t$ for n sufficiently large, where Z is standard normal. Now consider the actual particle location after n steps, assuming a constant plume velocity v. A randomly selected particle moves by an amount $v\Delta t + X_i$ over the time interval Δt, so after n jumps the particle is located at

$$X_1 + \cdots + X_n + vt \approx vt + \sqrt{Dt}Z.$$

The probability density function of $\sqrt{Dt}Z$ was given in equation (7.32) in Section 7.4. Since the term vt is not random, a simple change of variable shows that $vt + \sqrt{Dt}Z$ has probability density function

$$C(x,t) = \frac{1}{\sqrt{2\pi Dt}}\, e^{-(x-vt)^2/(2Dt)} \tag{9.17}$$

for any $t > 0$. This change of variable shifts the center of the graph from $x = 0$ to $x = vt$.

Next we will use Fourier transforms to see how (9.17) solves a diffusion equation with drift. Using (7.30) along with a change of variable $y = (x - vt)/\sqrt{Dt}$ yields

$$\hat{C}(k,t) = \int_{-\infty}^{\infty} e^{-ikx} C(x,t)\, dx = e^{-ikvt - Dtk^2/2}$$

when $C(x,t)$ is given by (9.17). Then

$$\frac{d\hat{C}}{dt} = -v(ik)\hat{C} + \frac{D}{2}(ik)^2\hat{C}, \tag{9.18}$$

and inverting the Fourier transform leads to

$$\frac{\partial C}{\partial t} = -v\frac{\partial C}{\partial x} + \frac{D}{2}\frac{\partial^2 C}{\partial x^2}, \tag{9.19}$$

the diffusion equation with drift.

To solve this equation by particle tracking, simulate a large number N of random particles, using a Monte Carlo simulation of $(X_1 + v\Delta t) + \cdots + (X_n + v\Delta t)$ with $t = n\Delta t$. Then a relative frequency

Algorithm: PARTICLE TRACKING CODE

Variables: N = number of particles
T = time of final particle jump
M = number of particle jumps
v = particle velocity
D = particle dispersivity
$t(j)$ = time of jth jump
$S(i, j)$ = location of ith particle at time $t(j)$

Input: N, T, M, v, D

Process:
Begin
$\Delta t \leftarrow T/M$
for $j = 0$ to M do
  Begin
  $t(j) \leftarrow j\Delta t$
  End
for $i = 1$ to N do
  Begin
  $S(i, 0) \leftarrow 0$
  for $j = 0$ to $M - 1$ do
    Begin
    $S(i, j + 1) \leftarrow S(i, j) + \text{Normal } (v\Delta t, \sqrt{D\Delta t})$
    End
End

Output:
$t(1), \ldots, t(M)$
$S(1, 1), \ldots, S(N, 1)$
$\vdots$
$S(1, M), \ldots, S(N, M)$

Figure 9.21: Pseudocode for particle tracking simulation of Section 7.5.

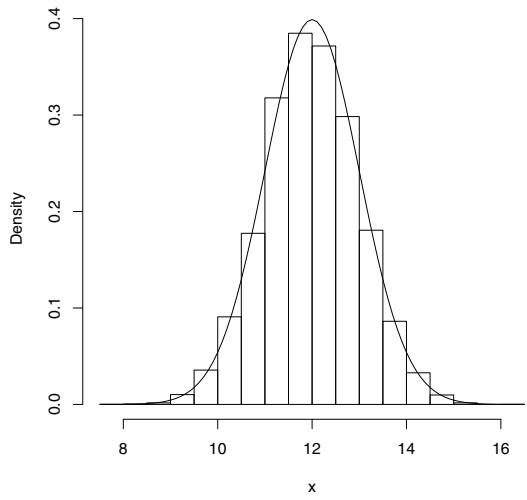


Figure 9.22: Relative frequency histogram showing results of the particle tracking code in Figure 9.21 with $N = 10,000$, $T = 4$, $M = 50$, $v = 3.0$, and $D = 0.25$. The corresponding analytical solution (solid line) to the diffusion equation with drift (9.19) at time $t = 4$ is also shown for comparison.

histogram of these particle locations will approximate the probability density curve $C(x,t)$ that gives the theoretical relative concentration for an infinite number of particles. Figure 9.21 shows the pseudocode for this algorithm. Figure 9.22 shows the results of a computer implementation, to solve the pollution problem of Example 7.5 with $v = 3.0$ km/hr and $D = 0.25$ km^2/hr. The simulation used $N = 10,000$ particles, a final time of $T = 4$ hours, and $M = 50$ time steps. Figure 9.22 also shows the analytical solution (9.17) for comparison. It is apparent that the histogram gives a reasonable approximation to the normal density curve. The particle model can be considered as the basic physical model for this problem. The density curve is an approximation, justified by the central limit theorem.

The particle tracking algorithm can easily be extended to allow the coefficients v, D in (9.19) to vary in space. When the velocity v varies in space, the only difference is that the mean jump size $v\Delta t$ for any given particle depends on the current location. In other words, the particle jumps become a Markov chain on the continuous state space $-\infty < x < \infty$ and the jump distribution depends on the current state in a very simple way. As $\Delta t \to 0$, the graph with these Markov chain jumps converges to a *diffusion process*, whose density function $C(x,t)$ solves the diffusion equation with drift (9.19), see Friedman (1975) for more details.

Now we continue with Step 3 of the pollution problem. Using the method of particle tracking, we will model the pollution plume as a cloud of random particles, and we will use Monte Carlo simulation to track the plume movements. We assume a coordinate system where the pollutant release occurred at location 0 and the town is at location 10 kilometers. Then the distance between a particle at location x and the center of town is given by $d = |x - 10|$. All N particles are located at $x = 0$ at time $t = 0$. At each time step Δt, each particle makes a random movement $v\Delta t + X_i$ where

$$v = v(x) = \begin{cases} 3 & \text{if } |x - 10| > 10, \\ 8 - 0.5|x - 10| & \text{if } |x - 10| \le 10. \end{cases} \tag{9.20}$$

Each X_i has mean 0 and variance $D\Delta t$ where $D = 0.25$. For our simulation, we will take each X_i to be normally distributed. We will also assign a mass to each particle. Since the maximum pollution concentration at time $t = 1$ hour is 20×50 ppm $= 1000$ ppm, the pollution concentration $P(x, t)$ is related to the relative concentration $C(x, t)$ by $P = P_0 C$ where $P_0 = 1000\sqrt{0.5\pi}$, using the calculations of Section 7.4. Hence we will assign each particle a concentration $\Delta P = P_0/N$. In our problem setup, $x = 0$ represents the pollution source, and $x = 10$ the center of town. We will estimate concentration in the center of town based on the number of particles at location $9.5 < x \le 10.5$. If K out of N particles lie in this interval, the relative concentration $C \approx K/N$, and the pollution level is $P \approx P_0(K/N) = K\Delta P$. If the time interval $[0, T]$ is divided into M jumps, then the time increment is $\Delta t = T/M$. The pseudocode for this particle tracking simulation is summarized in Figure 9.23. The variable $t(j)$ is the time after j times steps. The state variable $S(i, j)$ represents the location of particle i at time $t(j)$. The simulation only requires us to store the current state, i.e., we can replace $S(i, j)$ by S in the code. The variable $P(j)$ represents the estimated pollution level in town at time $t(j)$.

Step 4 is to implement the particle tracking code developed in Step 3. After implementing the algorithm in Figure 9.23, one run of the resulting code with final time $T = 4.0$ hours, $M = 100$ time steps, and $N = 1,000$ particles produced estimates of $P\,max = 397$, $T\,max = 1.96$, and $Tsafe = 2.32$ for our three MOP. Figure 9.24 shows the simulated concentration levels $P(j)$ at each time $t(j)$. The roughness in the graph is due to the particle tracking simulation. Using more particles (i.e., increasing N) would result in a smoother graph.

Step 5 is to answer the question. We estimate that the maximum pollution level in town will occur about 2 hours after the accident. The maximum pollution level in town is predicted to reach about 400 parts per million (ppm) by volume, which is about 8 times greater than the maximum safe level of this Category 1 contaminant. It is estimated that the pollution plume will continue through town, carried by the wind, and after 2 hours and 40 minutes, the toxic level in town will fall back down below the safety threshold. These estimates are based on a particle tracking simulation that approximates the path of pollution particles headed toward the town. The simulation takes variations in wind speed into account, since wind speeds are generally greater in the city. Figure 9.24

Algorithm: PARTICLE TRACKING CODE (Example 9.5)

Variables:
N = number of particles
T = time of final particle jump (hrs)
M = number of particle jumps
$t(j)$ = time of jth jump (hrs)
$P(j)$ = pollution concentration in town after j jumps (ppm)
$P\,max$ = maximum pollution concentration in town (ppm)
$T\,max$ = time of maximum pollution in town (hrs)
$Tsafe$ = time pollution in town falls to safe level (hrs)

Input: N, T, M

Process:
Begin
$\Delta t \leftarrow T/M$
$\Delta P \leftarrow 1000\sqrt{0.5\pi}/N$
for $j = 1$ to M do
  $t(j) \leftarrow j\Delta t$
  $P(j) \leftarrow 0$
for $i = 1$ to N do
  $S(i, 0) \leftarrow 0$
  for $j = 1$ to M do
    $v \leftarrow 3$
    if $|S(i, j-1) - 10| \leq 10$ then $v \leftarrow 8 - 0.5|S(i, j-1) - 10|$
    $S(i, j) \leftarrow S(i, j-1) + \text{Normal}\ (v\Delta t,\ 0.5\sqrt{\Delta t})$
    if $9.5 < S(i, j) \leq 10.5$ then $P(j) = P(j) + \Delta P$
$T\,max \leftarrow 0$
$P\,max \leftarrow 0$
$Tsafe \leftarrow 0$
for $j = 1$ to M do
  if $P(j) > P\,max$ then
    $P\,max \leftarrow P(j)$
    $T\,max \leftarrow t(j)$
  if $P(j) > 50$ then $Tsafe = t(j) + \Delta t$
End

Output: $T\,max$, $P\,max$, $Tsafe$

Figure 9.23: Pseudocode for particle tracking simulation of the pollution problem with variable wind speed.

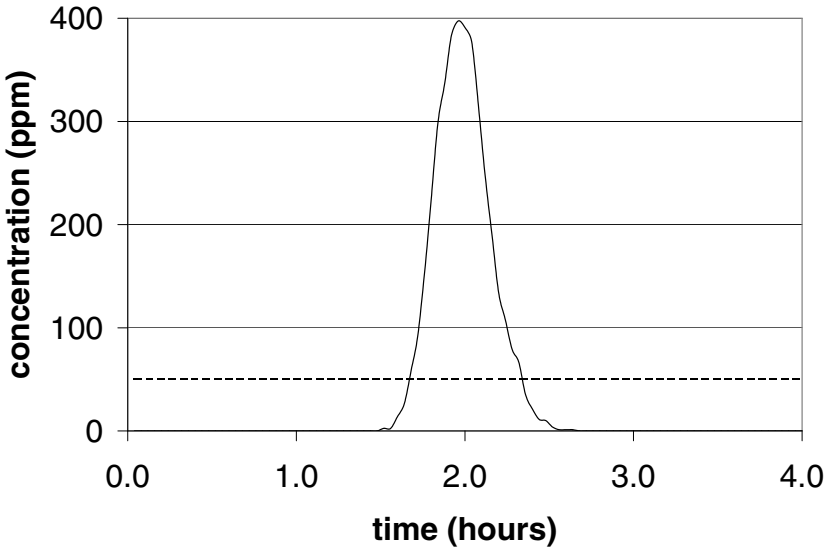


Figure 9.24: Estimated pollution concentration in center of town for the pollution problem with variable wind speed. The dotted line indicates the maximum safe level.

illustrates the predicted contamination levels at the center of town over time, starting at the time of the accident. The dotted line in that figure indicates the maximum safe level of 50 ppm.

For sensitivity analysis, we begin by investigating the sensitivity of our simulation results to random factors. Figure 9.25 illustrates the result of $R = 100$ simulation runs of the algorithm in Figure 9.23. Each simulation assumes a final time of $T = 4.0$ hours, using $M = 100$ time steps, and $N = 1,000$ particles. The average value of $P\,max$ for these simulations is around 400 ppm, with a typical range of ± 20 ppm. More specifically, the $R = 100$ simulation runs produced 100 values of $P\,max$, with sample mean 404.12 and sample standard deviation 16.18. The sample mean and sample standard deviation were computed using built-in functions in our programming platform. For R data points $X_1, \ldots, X_R$, the *sample mean* $\bar{x}$ is just the average,

$$\bar{x} = \frac{1}{R} \sum_{j=1}^{R} X_j.$$

The *sample variance* is given by the formula

$$s^2 = \frac{1}{R-1} \sum_{j=1}^{R} (X_j - \bar{x})^2,$$

which averages the squared deviations from the sample mean. Then the *sample*

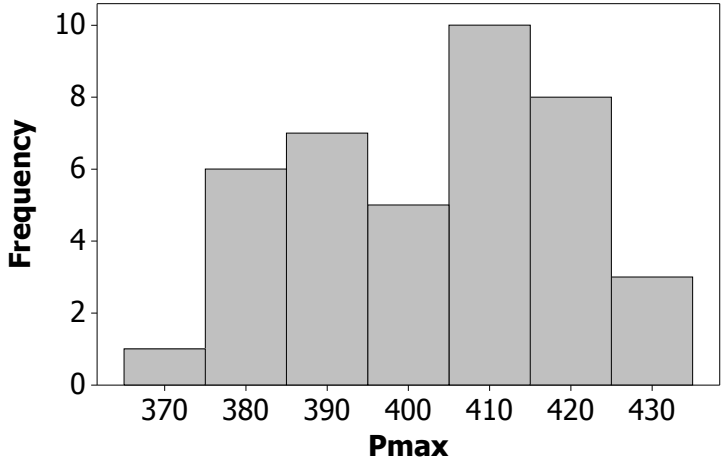


Figure 9.25: Estimated pollution concentration in center of town for the pollution problem with variable wind speed. Results of 100 simulations.

standard deviation s is the square root of the sample variance. Dividing by $R - 1$ instead of R in the formula for the sample variance ensures that the estimates of σ^2 are unbiased, i.e., repeated application of this formula in the same setting gives estimates that average to the correct value. The sample mean can be considered a typical value, and the sample standard deviation measures the typical spread of values. See Ross (1985) for more details.

The most common value of the time $T\,max$ of the maximum concentration was 1.92 hours, and every trial gave a value between 1.88 and 1.96. The most common value of the time $Tsafe$ until a safe level was 2.32 hours, and all values were between 2.28 and and 2.36. Since the time step was $T/M = 0.04$, the difference between the most common values and the largest or smallest value was a single time step. We conclude that the $T\,max$ and $Tsafe$ are relatively unaffected by random factors. The sensitivity of $P\,max$ to random factors is somewhat greater. This standard deviation depends on the number of particles N used in each simulation.

To test this, we conducted $R = 10$ additional simulations with $N = 10,000$ particles, using the same values of the remaining parameters. The resulting values for the maximum concentration $P\,max$ were all between 395 and 408, with a sample mean of 400 and a sample standard deviation of 5.6. A simple probability model explains how the variation in outcomes depends on the sample size. The estimated value of $P\,max$ is $K\Delta P$ where $\Delta P = P_0/N$ is not random, and K is the random number of particles that lie in the interval $9.5 < x \leq 10.5$.

The quantity $P\,max = P_0 q$ where

$$q = \int_{9.5}^{10.5} C(x,t)\,dx$$

is the theoretical probability that a given particle will lie in this interval, at time $t = T\,max$. As in Section 9.1, set $Y_i = 1$ if the ith particle lies in this interval, and $Y_i = 0$ otherwise, so that each Y_i has mean q and variance $\sigma^2 = q(1-q)$. Then $K = Y_1 + \cdots + Y_N$ is the number of particles that lie in this interval. The strong law of large numbers guarantees that

$$K\Delta P = P_0(K/N) = P_0 \frac{Y_1 + \cdots + Y_N}{N} \to P_0 q = P\,max$$

as the number of particles $N \to \infty$. The central limit theorem implies that K/N is unlikely to vary from its mean q by more than $2\sigma/\sqrt{N}$ for N large. Hence increasing the number of particles by a factor of 10 should decrease the variability of simulation results by a factor of $\sqrt{10} \approx 3$, which is consistent with our results. To go a bit further, since the total mass of all particles is

$$P_0 = \int_{-\infty}^{\infty} P(x,t)\,dx = 1000\sqrt{0.5\pi} \approx 1253 \text{ ppm},$$

the maximum concentration of 400 ppm represents about 30% of the particles. Then we could predict that around 68% of our simulated estimates for $P\,max$ should lie within $P_0\sqrt{(0.3)(0.7)/N}$ ppm of the true value, which evaluates to around 18 ppm for $N = 1000$ and 5.7 ppm for $N = 10,000$. This is reasonably consistent with the observed standard deviations (16 and 5.6, respectively) for those simulations. A similar analysis shows that variations in the estimated concentration curve Figure 9.24 decrease like the square root of the number of particles in the simulation. Hence a larger number of particles will result in a smoother curve. However, since the variations fall off like the square root of the sample size, it may not always be practical to simulate enough particles to get a completely smooth curve.

Next we consider the robustness of our results to the assumed form of the velocity function $v = v(x)$. A comparison with the results of Section 7.4 shows that, compared to the case of a constant velocity, the particle tracking model predicts a faster arrival of the peak concentration, and a lower level of that concentration. Since the safe level is 50 ppm, the highest concentration in town is $50 \times 11 = 550$ ppm at a constant wind speed of 3.0 km/hr, arriving after 3.3 hours. This can also be checked using a simple modification of the particle tracking code in Figure 9.23, by deleting the line:

$$\text{if } |S(i,j) - 10| \le 10 \text{ then } v \leftarrow 8 - 0.5|S(i,j) - 10|.$$

Then the velocity is fixed at 3.0 km/hr throughout the simulation. One simulation run of the modified code with final time $T = 6.0$ hours, $M = 150$ time steps, and $N = 10,000$ particles produced estimates of $P\,max = 528$ ppm at

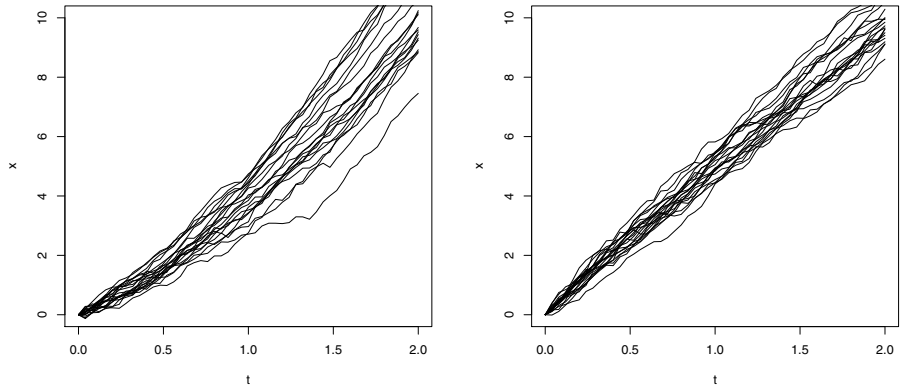


Figure 9.26: Left panel: Particle traces for particle tracking model with variable wind speed (9.20). Right panel: Particle traces for particle tracking model with constant wind speed $v = 5$ km/hr.

time $Tmax = 3.28$ hours, and $Tsafe = 4.04$ hours, in good agreement with the results from Section 7.4. In summary, it seems that the variation in wind speed causes the peak to arrive sooner, with a lower concentration. Is this due to the fact that the "average" wind speed is increasing?

The sensitivity analysis in Table 7.1 shows that the peak concentration in town ranges between $50 \times 11 = 315$ ppm at a wind speed of 1.0 km/hr to $50 \times 14.2 = 710$ ppm at a wind speed of 5.0 km/hr. That is, a higher wind speed leads to higher peak concentration in town, arriving earlier. This makes sense, because at a higher wind speed, the plume arrives sooner, with less time to spread. Indeed, the particle tracking simulation with the variable wind speed $v(x) \geq 3$ km/hour given by (9.20) predicts that the plume peak will arrive sooner. However, the model also predicts a *lower* peak concentration. To understand this paradox, we will take a closer look at the individual particle traces.

The left panel in Figure 9.26 graphs the paths of $N = 20$ particles from the particle tracking model with variable wind speed (9.20). These graphs were produced using the algorithm in Figure 9.23 by plotting $S(i, j)$ versus $t(j)$ for $j = 1, 2, \ldots, M$ for particles $i = 1, 2, \ldots, N$. Compare the right panel in Figure 9.26, which graphs the paths of $N = 20$ particles with constant wind speed $v = 5$, chosen so that both plumes reach the city, located at coordinate $x = 10$, at around the same time. Compared to the constant velocity model, the variable velocity paths are significantly more dispersed. A closer inspection of the left panel shows the reason for this dispersion. Because of random factors, some particle begin more slowly, and therefore experience lower velocities in the subsequent jumps. Other particles begin faster, and remain on a faster path. The term *dispersion* is used to describe the spreading of particles due to

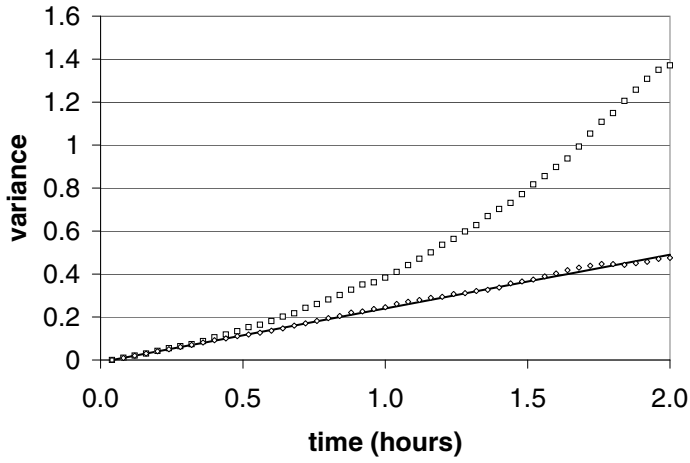


Figure 9.27: Observed variance for particle tracking model (boxes) with variable wind speed (9.20) compared to observed variance for the same model with constant wind speed $v = 5$ km/hr (diamonds), and theoretical variance for the model with constant wind speed (solid line).

differences in velocity. For pollution problems in air and water, dispersion is often the main driving factor behind plume spreading.

In engineering models of contaminant transport, it is common to use the diffusion equation with drift (9.19) as the basic model for plume evolution. It has been noted in many studies that the parameter D, determined by fitting a Gaussian probability density curve to data, tends to grow with time (or with distance traveled by the plume center of mass). This super-diffusion is often attributed to variations in the particle velocities. An interesting fractal model for porous media, used in ground water contamination studies, has been proposed to explain the observation that D tends to grow like a power law, $D = D_0 t^p$ for some $p > 0$, in many studies. See Wheatcraft and Tyler (1988) for more details.

Figure 9.27 documents super-dispersion in the particle tracking model with variable velocity. The graph shows the results of a particle tracking simulation with $N = 1000$ particles with final time $T = 2.0$ hours, and $M = 50$ time steps, so that the time increment $t\Delta t = 0.04$ is the same as before. The boxes shows the variance of the particle locations $\{S(i, j) : 1 \le i \le N\}$ at each time $t(j)$ for $j = 1, \ldots, M$, using the algorithm in Figure 9.23. For a constant velocity, the variance grows linearly, since the particle location at time t is $vt + Z_t$, where Z_t is normally distributed with mean zero and variance $= Dt$. This is the theoretical variance line on the graph. The results of another particle tracking simulation, using the algorithm in Figure 9.23 with constant velocity $v = 5$ km/hr, and all other parameters the same, was run to check this theoretical result. The diamonds on the graph show the resulting variance in particle

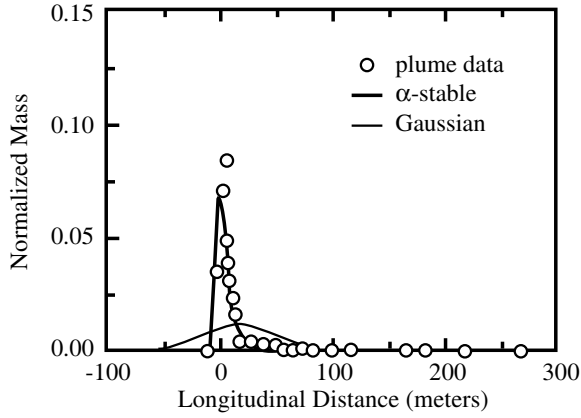
locations at each time step, to demonstrate how closely the observed variance matches the theoretical model. Note that a constant velocity simulation with $v = 3$ or $v = 8$ or any other value would produce the same result, since the constant velocity only changes the mean particle location, and has no effect on the variance.

The super-dispersion in contaminant plumes creates a significant problem in practice, since it means that the constant coefficient equation (9.19) is inadequate to predict plume behavior. A variety of methods have been applied to address this problem, including more complicated systems of differential equations in one or more dimensions with variable coefficients, that can be solved by particle tracking or finite difference codes similar to those developed in Chapter 6 of this book. Recently, an interesting new idea has been proposed to model super-diffusive plumes using fractional calculus. The new model replaces the second derivative in (9.19) by a fractional derivative of order $1 < \alpha < 2$. In the next section, we will investigate an application of this fractional diffusion model to ground water pollution.

9.5 Fractional Diffusion

Fractional derivatives were invented by Leibnitz in 1695, soon after their integer order cousins, but have only recently found practical applications. Now they are used in models of anomalous diffusion in water and air pollution, heat transfer in complex materials, invasive species, cell membranes, and electronics. In this section, we will use particle tracking to explore a fractional calculus model for water pollution.

Example 9.6. An experiment to study dispersion in ground water was conducted on a United States Air Force base in Columbus, Mississippi in 1993 (see Boggs et al. (1993) for further details). Water containing a radioactive tracer (tritium) was injected underground, and the movement of the tritium plume through the ground water was tracked over time. Figure 9.28 shows the relative concentration $C(x, t)$ of tritium tracer versus distance x meters downstream from the injection site, based on measurements taken $t = 224$ days after injection. The thin curve near the bottom of the graph is the best fitting Gaussian density, the solution to the diffusion equation with drift (9.19). The movement of the center of mass over the course of the experiment was roughly proportional to elapsed time. The plume spread proportional to $t^{0.9}$, evidence of anomalous super-dispersion. It is also clear that the plume shape is strongly skewed in the positive (downstream) direction, so that the symmetric Gaussian solution to the traditional diffusion equation (9.19) cannot provide an adequate model for this data. Tritium levels were be measured in Curies, a unit of radioactivity. One Curie (Ci) is defined as 3.7×10^{10} decays per second. In this experiment, $P_0 = 540$ Ci of tritium were injected. Estimate the time and amount of the maximum tritium level at a point $x = 20$ meters downstream of the injection site, as well as the time required for the tritium concentration at that location to fall below 2 Ci.



c) Snapshot 3 (day #224)

Figure 9.28: Measured tritium concentration (hollow circles) from the fractional diffusion problem of Example 9.6 with fitted stable density (thick solid line) and fitted Gaussian density (thin black line), from Benson et al. (2001).

We will use the five-step method. The first step is to ask a question. We adopt a coordinate system where $x = 0$ is the injection site, and water flows in the positive x direction, consistent with Figure 9.28. We want to predict the concentration of tritium tracer at location $x = 20$ meters downstream from the injection site, and how it varies over time. The results of step 1 are summarized in Figure 9.29.

Step 2 is to select the modeling approach. We will use a fractional diffusion model, and solve the model by the method of particle tracking.

The *fractional derivative* $\partial^\alpha C/\partial x^\alpha$ can be defined as the function with Fourier transform $(ik)^\alpha \hat{C}$. The fractional diffusion equation with drift is

$$\frac{\partial C}{\partial t} = -v\frac{\partial C}{\partial x} + D\frac{\partial^\alpha C}{\partial x^\alpha} \tag{9.21}$$

where $1 < \alpha < 2$. Take the Fourier transform on both sides to get

$$\frac{d\hat{C}}{dt} = -v(ik)\hat{C} + D(ik)^\alpha \hat{C}$$

and solve, using the point source initial condition $\hat{C}(k, 0) \equiv 1$, to arrive at

$$\hat{C}(k,t) = \int_{-\infty}^{\infty} e^{-ikx} C(x,t)\, dx = e^{-ikvt + Dt(ik)^\alpha} \tag{9.22}$$

for any $t > 0$. This Fourier transform cannot be inverted in closed form, but it is known from probability theory that $C(x, t)$ is an α-*stable* density function.

Variables:	t = Time since release of tracer (days) x = Distance downstream from injection site (m) P = Pollution concentration at location x at time t (Ci)
Assumptions:	Average plume velocity is constant Plume spreads proportional to $t^{0.9}$ Plume is positively skewed
Objective:	Determine the maximum tracer concentration 20 m downstream, time the maximum occurs, and time until concentration falls below 2 Ci.

Figure 9.29: Results of Step 1 of the water pollution problem.

The stable density appears in an *extended central limit theorem*. Suppose that $X, X_1, X_2, \ldots$ are independent and identically distributed random variables. If $\Pr\{X > x\} = Ax^{-\alpha}$ for some $A > 0$ and some $1 < \alpha < 2$, then the extended central limit theorem implies that

$$\lim_{n\to\infty} \Pr\left\{ \frac{X_1 + \cdots + X_n - n\mu}{n^{1/\alpha}} \le x \right\} \to \Pr\{Z_\alpha \le x\} \tag{9.23}$$

where $\mu = E(X)$, and Z_α is a stable random variable with index α, whose density $g_\alpha(x)$ has Fourier transform

$$\hat{g}(k) = \int_{-\infty}^{\infty} e^{-ikx} g(x)\, dx = e^{D(ik)^\alpha} \tag{9.24}$$

for some $D > 0$ depending on A and α. Because $\sigma^2 = V(X) = \infty$ in this case, the usual central limit theorem, discussed in Section 7.3, does not apply here. See Meerschaert and Sikorskii (2012) for more details about stable laws, the extended central limit theorem, and fractional derivatives.

The spreading rate of the solution $C(x,t)$ to the fractional diffusion equation can be determined from the Fourier transform. A simple change of variables shows that the mean-centered concentration $C_0(x,t) = C(x + vt, t)$ has Fourier transform

$$\int_{-\infty}^{\infty} e^{-ikx} C_0(x,t)\, dx = e^{Dt(ik)^\alpha}, \tag{9.25}$$

and another change of variables shows that $t^{-1/\alpha} C_0(t^{-1/\alpha}x, 1)$ has the same Fourier transform as $C_0(x,t)$. This shows that the peak concentration falls like $t^{1/\alpha}$, and the plume spreads like $t^{1/\alpha}$ away from its center of mass at $x = vt$. Since $\alpha < 2$, this means that the

plume spreads faster than a traditional diffusion, so that the fractional diffusion equation with drift (9.21) is a model for anomalous super-diffusion. Finally, note that in the case $\alpha = 2$, the fractional diffusion model reduces to traditional diffusion.

Step 3 is to formulate the model. We will model the tritium plume using the fractional diffusion equation (9.21). We will solve this equation by the method of particle tracking introduced in Section 9.4, using the extended central limit theorem (9.23). The random variables $X_1, \ldots, X_n$ in (9.23) are identically distributed with a random variable X with cumulative distribution function

$$F(x) = \begin{cases} 0 & \text{for } x < A^{1/\alpha} \\ 1 - Ax^{-\alpha} & \text{for } x \geq A^{1/\alpha}. \end{cases} \tag{9.26}$$

Its probability density function is

$$f(x) = \begin{cases} 0 & \text{for } x < A^{1/\alpha} \\ A\alpha x^{-\alpha-1} & \text{for } x \geq A^{1/\alpha}, \end{cases}$$

and its mean is

$$\mu = \int_{A^{1/\alpha}}^{\infty} x\, A\alpha x^{-\alpha-1} dx = A^{1/\alpha} \frac{\alpha}{\alpha - 1}.$$

To simulate the random variable X, we will use the inverse cumulative distribution method developed in Example 9.3. Set $y = F(x) = 1 - Ax^{-\alpha}$ and invert to get

$$x = F^{-1}(y) = \left(\frac{A}{1-y} \right)^{1/\alpha}$$

and then take $X = F^{-1}(U)$, where U is uniform on $(0, 1)$. To confirm that X has the desired distribution, write

$$\begin{aligned} \Pr\{X \leq x\} &= \Pr\left\{ \left(\frac{A}{1-U} \right)^{1/\alpha} \leq x \right\} \\ &= \Pr\left\{ U \leq 1 - Ax^{-\alpha} \right\} = 1 - Ax^{-\alpha} \end{aligned}$$

when $x \geq A^{1/\alpha}$, so that $0 < 1 - Ax^{-\alpha} < 1$.

For the particle tracking model, assume that each particle makes a small random movement $v\Delta t + (\Delta t)^{1/\alpha} X_i$ over a small time interval $\Delta t = t/n$, where

$$X_i = \left(\frac{A}{1-U_i} \right)^{1/\alpha} - A^{1/\alpha} \frac{\alpha}{\alpha - 1}$$

and $U_1, \ldots, U_n$ are independent uniform $(0, 1)$ random variables. Use the extended central limit theorem (9.23) with $\mu = 0$ to see that $X_1 + \cdots + X_n \approx n^{1/\alpha} Z_\alpha$ for large n. Since $t = n\Delta t$, we have

$$\Big(v\Delta t + (\Delta t)^{1/\alpha} X_1 \Big) + \cdots + \Big(v\Delta t + (\Delta t)^{1/\alpha} X_n \Big) \approx vt + (n\Delta t)^{1/\alpha} Z_\alpha = vt + t^{1/\alpha} Z_\alpha.$$

The distribution function of Z_α is

$$G(x) = \int_{-\infty}^{x} g(u)\,du$$

and so the distribution function of the limit $vt + t^{1/\alpha}Z_\alpha$ is

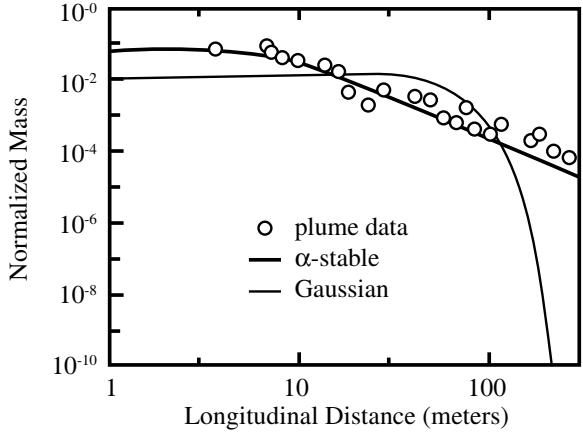
$$\begin{aligned}\Pr\{vt + t^{1/\alpha}Z_\alpha \le x\} &= \Pr\left\{Z_\alpha \le \frac{x - vt}{t^{1/\alpha}}\right\} \\ &= \int_{-\infty}^{t^{-1/\alpha}(x-vt)} g_\alpha(u)\,du \\ &= \int_{-\infty}^{x} g_\alpha(t^{-1/\alpha}(y - vt))t^{-1/\alpha}dy\end{aligned}$$

using the substitution $u = t^{-1/\alpha}(y - vt)$ in the last line. Then the density of $vt + t^{1/\alpha}Z_\alpha$ has Fourier transform

$$\begin{aligned}\hat{C}(k,t) &= \int_{-\infty}^{\infty} e^{-ikx} g(t^{-1/\alpha}(y - vt))t^{-1/\alpha}dy \\ &= \int_{-\infty}^{\infty} e^{-ik(vt+t^{1/\alpha}u)} g_\alpha(u)\,du \\ &= e^{-ikvt}\int_{-\infty}^{\infty} e^{-i(kt^{1/\alpha})u} g_\alpha(u)\,du \\ &= e^{-ikvt+Dt(ik)^\alpha}\end{aligned}$$

using the same substitution $u = t^{-1/\alpha}(y - vt)$ again, along with (9.24). Since this is identical to (9.22), it follows that $C(x,t)$ solves the fractional diffusion equation with drift (9.21). Hence a relative frequency histogram of particle locations will approximate the probability density curve $C(x,t)$ that gives the theoretical relative concentration for an infinite number of particles.

The α-stable curve in Figure 9.28 was obtained by numerically inverting the Fourier transform (9.22) for the stable density, using a fitted velocity of $v = 0.12$ m/day. The robustness of this velocity assumption will be discussed later in this section. The stable index $\alpha = 1.1$ was chosen to give a spreading rate of $t^{1/1.1} \approx t^{0.9}$. The fitted value of α was also verified by checking the rate of peak concentration decline, and the leading tail of the concentration curve. Figure 9.30 shows the same tracer data from Figure 9.28 on a log-log scale, along with the fitted Gaussian and stable models. The α-stable density $C = C(x,t)$ has the property that $C \approx t\alpha A x^{-\alpha-1}$ for x sufficiently large, so that $\log C \approx \log(t\alpha A) - (\alpha+1)\log x$. Hence a log-log plot of the fitted stable density will appear linear for large x. (The slope of this line can be used to estimate the parameter α.) The fact that the concentration data also follows this line provides additional evidence in favor of the fractional diffusion model. Since the normal estimate of tracer concentration is too low by a factor of 10^6 at the plume leading edge, the traditional diffusion model seriously under-estimates the risk



a) Snapshot 3 (day #224)

Figure 9.30: Tracer data and fitted models from Figure 9.28 on a log-log scale, to show the power law leading tail.

of downstream contamination. The fractional dispersivity $D = 0.14$ m$^\alpha$/day in (9.21) was chosen to give the best fitting stable density curve for all time. (Data was collected at day 27, 132, 224, and 328). This lead to an estimate $A = 0.0131$ for the particle jumps, using the formula $A = D(\alpha - 1)/\Gamma(2 - \alpha)$ from Theorem 3.41 in Meerschaert and Sikorskii (2012). The gamma function is discussed in Exercise 23 at the end of this chapter.

The code for our particle tracking simulation is listed in Figure 9.31. The code is similar to Figure 9.23, with a few modifications. The total mass of the particles is $P_0 = 540$ Ci, and the target concentration is 2 Ci. The tritium concentration at location $x = 20$ is estimated by counting the number of particles in the interval $15 < x \leq 25$, and then dividing by the length of this interval. The two distribution parameters are also listed. The parameter α governs the tail, and then A determines the scale, e.g., the mean value is proportional to $A^{1/\alpha}$. Since $P_0 = 540$ Ci of tritium were injected, we model the tritium concentration using $P(x, t) = P_0 C(x, t)$.

Before proceeding to Step 4, we tested the code in Figure 9.31, to see if we could reproduce the concentration curve in Figure 9.28. Figure 9.32 shows a relative frequency histogram of the locations $S(i, M)$ of particles $i = 1, \ldots, N$ at time $T = t(M) = 224$ days from a particle tracking simulation with $N = 10,000$ particles and $M = 100$ time steps, based on a computer implementation of the code in Figure 9.31. We are satisfied that the histogram of particle locations in Figure 9.32 matches the α-stable density curve in Figure 9.28 reasonably well.

Step 4 is to solve the model. Figure 9.33 summarizes the results of running the code in Figure 9.31 with $N = 10,000$ particles, with a final time of $T = 4000$ days and $M = 100$ time steps. The graph could be made smoother by increasing

Algorithm: PARTICLE TRACKING CODE (Example 9.6)

Variables:
N = number of particles
T = time of final particle jump (days)
M = number of particle jumps
$t(j)$ = time of jth jump (days)
$P(j)$ = tritium concentration after j jumps (Ci)
$P\,max$ = maximum tritium concentration (Ci)
$T\,max$ = time of maximum concentration (hrs)
$Tsafe$ = time concentration falls to 2.0 Ci (hrs)
$A = 0.0131$ scale parameter in Eq. (9.26)
$\alpha = 1.1$ tail parameter in Eq. (9.26)

Input: N, T, M

Process:
Begin
$\Delta t \leftarrow T/M$
$\Delta P \leftarrow 540/N$
for $j = 1$ to M do
  $t(j) \leftarrow j\Delta t$
  $P(j) \leftarrow 0$
for $i = 1$ to N do
  $S(i, 0) \leftarrow 0$
  for $j = 1$ to M do
    $U \leftarrow \text{Random}(0, 1)$
    $X \leftarrow (A/(1-U))^{1/\alpha} - A^{1/\alpha}\alpha/(\alpha - 1)$
    $S(i, j) \leftarrow S(i, j-1) + 0.12\Delta t + \Delta t^{1/\alpha} X$
    if $15 < S(i, j) \leq 25$ then $P(j) = P(j) + \Delta P/10$
$T\,max \leftarrow 0$
$P\,max \leftarrow 0$
$Tsafe \leftarrow 0$
for $j = 1$ to M do
  if $P(j) > P\,max$ then
    $P\,max \leftarrow P(j)$
    $T\,max \leftarrow t(j)$
  if $P(j) > 2.0$ then $Tsafe = t(j) + \Delta t$
End

Output: $T\,max$, $P\,max$, $Tsafe$

Figure 9.31: Pseudocode for particle tracking simulation of the fractional diffusion problem in Example 9.6.

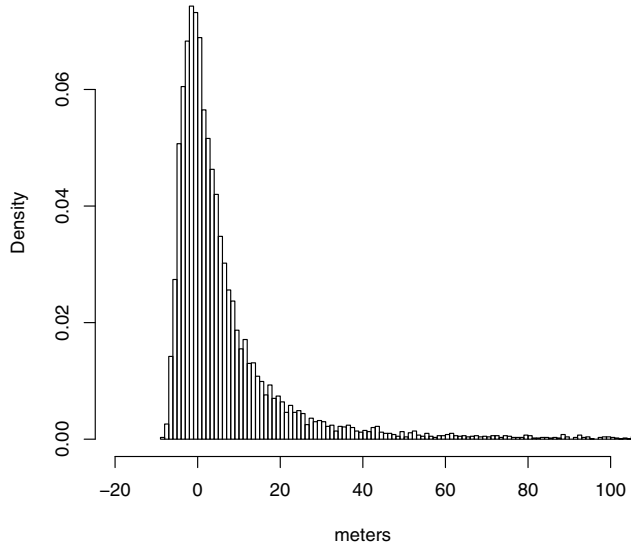


Figure 9.32: Results of particle tracking simulation for the fractional diffusion problem of Example 9.6 at time $t = 224$ days. The histogram approximates the solution $C(x, t)$ to the fractional diffusion equation with drift (9.21).

the number of particles. The maximum concentration of $Pmax = 9.14$ Ci occurred at time $Tmax = 1080$ days. It took $Tsafe = 2840$ days for the tritium level to fall back below 2 Ci.

Step 5 is to answer the question. We estimate that the maximum concentration of tritium at a point $x = 20$ meters downstream of the injection site will occur in approximately 3 years. At that time, the estimated concentration will be around 9 Curies. It will take almost 8 years for the tritium concentration at this point to fall back below 2 Curies, as the tritium plumes is carried downstream by the underground water. Figure 9.33 illustrates the predicted tritium concentration levels over time at the point $x = 20$ meters downstream, starting at the time of injection. The dotted line in that figure indicates the nominal level of 2 Curies. It is also relevant to note that the predicted tritium plume is stretched out in the direction of the ground water flow. Figure 9.32 illustrates the predicted plume shape 224 days after injection.

Since the results reported in Step 5 come from a Monte Carlo simulation, it is important to explore the sensitivity to random factors. We repeated the simulation 30 times with $N = 1,000$ particles, with a final time of $T = 4000$ days and $M = 100$ time steps. These are the same values as in Step 4, except that we decreased the number of particles per run, in order to speed up the simulation. The resulting values of the three MOP are $Pmax = 9.6 \pm 0.6$, $Tmax = 944 \pm 125$, and $Tsafe = 2830 \pm 110$, where we report the sample mean

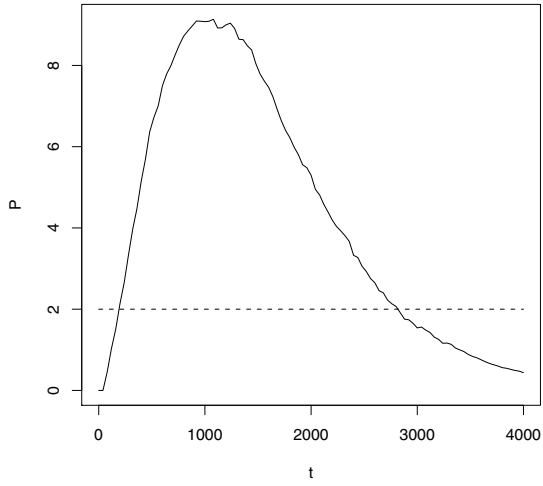


Figure 9.33: Estimated tritium levels at location $x = 20$ meters downstream of the injection point, for the water pollution problem of Example 9.6.

$\pm$ the sample standard deviation. Since the mean values from our 30 simulations are consistent with the values reported in Step 5, and the standard deviations are less than one unit (Curies or years), we conclude that the values reported in Step 5 are reliable, to the accuracy stated there.

Next we want to consider the robustness of the estimation procedure for the mean particle velocity. Recall that the velocity $v = 0.12$ m/day was estimated based on the mean (weighted average) of concentration measurements. Since the concentration curve is highly skewed, the mean (plume center of mass) is far from the mode (point of highest concentration). For example, at time $t = 224$ days, the mean is $x = vt = (0.12)(224) = 26.88$ meters. However, the peak of the α-stable density curve in Figure 9.28 lies far to the left of this point. The mean deviates significantly from the mode, because the concentration curve has a *heavy tail*. Consider the histogram in Figure 9.32. For the $N = 10,000$ particles simulated, the sample mean location was 18.5 meters, and the sample standard deviation was 355.2 meters in that Monte Carlo simulation. This absurdly large sample standard deviation is due to the fact that the theoretical standard deviation is infinite (since the second moment does not exist). Hence the sample standard deviation provides no useful information, other than to highlight the fact that there is a considerable spread in the particle location data. In fact, particle locations ranged from -8.45 to 28,352.3 meters. A very small number of particles travel a very long distance. Recall that the probability of jumping more than r meters downstream falls off proportional to $r^{-\alpha}$ in our particle jump model. Since $\alpha \approx 1$, this means that about 1 out of 10,000 particles will jump 10,000 times farther than usual. In statistics, these extreme data points

are called *outliers*.

In the presence of outliers, the mean can be an unreliable estimator of typical behavior. Consider a company in which the owner nets 1,000,000 per year, and each of 20 employees makes 50,000 per year. The average salary, including the owner, is $(2,000,000)/21 \approx 95,000$ per year, but this is hardly a good indicator of a typical salary. Likewise, the average particle location in the tritium plume is not a reliable indicator of plume center, because a small number of outliers inflate the average. The median of particle location is around 2.0 meters (50% of particles travel further downstream). The median is a more reliable measure of center than either the mean or the peak (mode), since it is not affected by outliers. However, there is no simple way to relate the median to the parameters in the model (9.21).

Note that the concentration readings in Figure 9.28 were all taken within approximately 300 meters of the injection site. This measurement procedure truncates the concentration plume. What affect does this have on the estimated velocity? Recall that the $N = 10,000$ particle locations from Figure 9.28 had a sample mean of 18.5 meters, which is not too far from the theoretical mean of 26.88 meters (considering the huge sample standard deviation). We sorted this location data, and found that 58 out of 10,000 data points exceeded 300 meters. Then we recomputed the mean particle location, omitting these values from the calculation (i.e., we added up the remaining data values, and divided by 9942). The resulting mean particle location was 8.33 meters, less than half the overall average of 18.5 meters. Although we used 99.4% of all the data values, the remaining 0.6% have a profound effect on the mean particle location. Since this calculation was based on a Monte Carlo simulation, we repeated this process several more times. The resulting mean location for all particles varied significantly. The standard deviation, and the location of the largest particle, varied widely. However, the mean location of those particles that end up less than 300 m downstream was always around 8 m. When the velocity parameter v was estimated from the concentration data in Figure 9.28 (and three other snapshots), this truncation effect was taken into account. In fact, the mean value estimated from the data in Figure 9.28 is 8.0 meters. The Gaussian curve in Figure 9.28 has a mean of approximately 8 m, since the truncation effect is negligible for this normal probability density function.

In modeling, the estimated parameter values in a given model must never be confused with the actual "truth" about these parameters. While both the traditional and fractional diffusion models contain velocity parameters, the meaning of those parameters can vary in the context of an application. For this reason, it would be a mistake to estimate the velocity parameter in the context of one model (traditional diffusion) and assume the same value is valid for a different model (fractional diffusion). The same applies to all model parameters. For example, it is not appropriate to estimate the tail parameter α in the power law model (9.26), and then assume that the same parameter value pertains to the stable model. In the finance literature, this simple truth has caused some serious confusion, see McCulloch (1997) for further details.

We conclude this example with a few comments about the connection be-

tween fractional derivatives, power law jumps, and fractals. We have already seen that the fractional diffusion equation, with a fractional derivative of order α, governs the sum of random power law jumps, where the number α is also the tail index of the power law jump probability. The probability model with power law jumps gives a concrete understanding of the fractional derivative. A good way to think about what the fractional derivative means is to consider the particle tracking model for a diffusion. The second derivative codes particle jumps with mean zero and finite variance. The fractional derivative codes power law jumps.

In Section 7.4, we derived the traditional diffusion equation (7.28) by applying the conservation of mass equation (7.26) along with Fick's Law (7.27). Recall that Fick's Law is empirical, i.e., it is based on observations of real world data in certain controlled experimental settings. The fractional diffusion equation can be derived in a similar manner, using a fractional Fick's law. In the fractional Fick's Law, the particle flux

$$q = -D\frac{\partial^{\alpha-1}C}{\partial x^{\alpha-1}} \tag{9.27}$$

is based on empirical observations from chaotic dynamical systems. There the proportion of particles that jump across j boxes of size Δx during one time step Δt falls off like a power law. This deterministic model for fractional diffusion is closely connected to the particle tracking model, based on random jumps with a power law distribution. See Meerschaert and Sikorskii (2012) for more details.

In Section 6.4 we discussed the interesting subject of fractals. Particles traces in both traditional and fractional diffusion are also random fractals. The right panel in Figure 9.26 shows some representative particle traces, obtained by plotting $S(n)$ versus $t(n)$ where

$$S(n) = \sum_{j=1}^{n} \left(v\Delta t + \sqrt{D\Delta t}\ Z_j \right), \tag{9.28}$$

and Z_j are independent standard normal random variables with mean zero and variance 1. The sum $S(n)$ has a normal distribution with mean vt and variance Dt, where $t = n\Delta t$. As $n \to \infty$, the discrete time stochastic process with value $S(n)$ at time $t(n)$ converges to a *Brownian motion* with drift $B(t) + vt$. The Brownian motion $B(t)$ is a Markov process whose state space is the entire real line. Its density functions $C(x,t)$ solve the diffusion equation (7.28). Particle traces of a Brownian motion are random fractals with dimension $d = 3/2$. If we replace the normal random variables in (9.28) with any other random variables with the same mean and variance, the central limit theorem still applies, and we get the same Brownian motion process in the limit. In two or more dimensions, the particle traces of a Brownian motion are random fractals with dimension $d = 2$. Although the dimension is an integer, these particle trajectories are still fractals, since their dimension is not equal to one.

Figure 9.34 shows a typical particle trace of Brownian motion. Since the particle trace is a fractal, zooming in on one portion of the graph will reveal

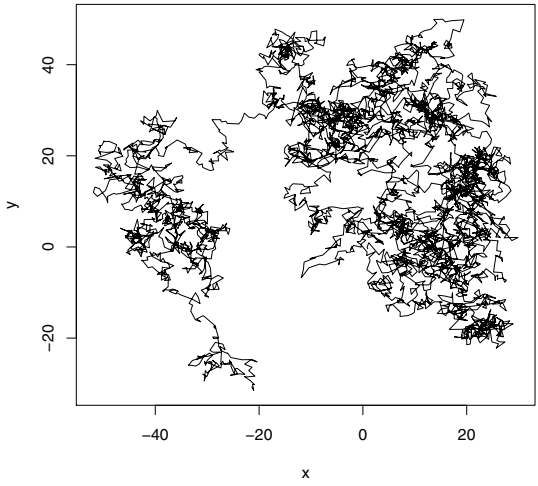


Figure 9.34: Particle tracking simulation of Brownian motion, a random fractal of dimension $d = 2$.

additional structure, similar to the larger picture. This is related to the self-similarity property of Brownian motion: The process $B(ct)$ at time scale $c > 0$ is probabilistically identical to the process $c^{1/2}B(t)$. Decreasing the time scale c decreases the spatial scale $c^{1/2}$, which is equivalent to zooming in on the graph.

If we replace the normal random variables in (9.28) with power law jumps that follow distribution (9.26) with $1 < \alpha < 2$, the extended central limit theorem shows that the limit is stable with index α. As $n \to \infty$, the discrete time stochastic process with value $S(n)$ at time $t(n)$ converges to a *stable Lévy motion with drift* $L(t) + vt$. The stable Lévy motion $L(t)$ is a close cousin of the Brownian motion. Its density functions $C(x, t)$ solve the fractional diffusion equation (9.21) with $v = 0$. In two or more dimensions, the particle traces of a α-stable Lévy motion are random fractals with dimension $d = \alpha$. Figure 9.35 shows a typical particle trace of stable Lévy motion, with $\alpha = 1.8$ in this case. The picture is similar to Brownian motion, except for occasional large jumps. The behavior is somewhat similar to the weather problem of Example 6.6. The trajectory remains localized for a period of time, and then jumps quickly to a different neighborhood. The process $L(ct)$ at time scale $c > 0$ is probabilistically identical to the process $c^{1/\alpha}L(t)$. As the fractal dimension α decreases, the graph becomes smoother. In summary, the parameter α codes the power law jumps, the order of the fractional derivative, and the fractal dimension of the particle traces.

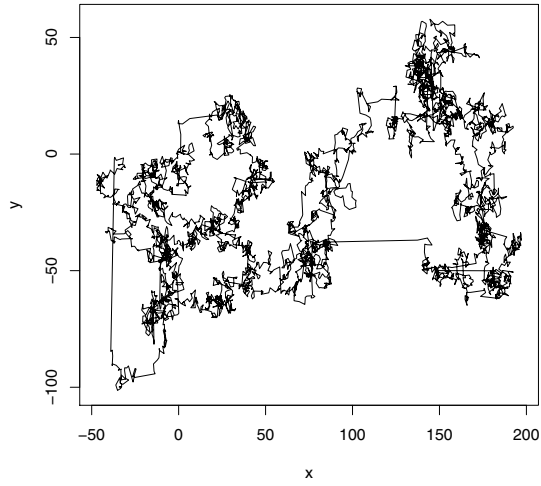


Figure 9.35: Particle tracking simulation of stable Lévy motion, a random fractal of dimension $d = \alpha = 1.8$.

9.6 Exercises

1. A simple game of chance is played by flipping a coin. The house flips a coin and the player calls it in the air. If the coin lands the way the player called it, the house pays the player \$1; otherwise, the player pays the house \$1. The player begins with \$10.

   (a) What are the odds that the player will go broke before he doubles his money? Use the five-step method, and model using a Monte Carlo simulation.

   (b) How long on average does the game described in part (a) last?

   (c) How much on average does the player have after 25 coin flips?

2. On a roll of two dice, a total of seven occurs with probability 1/6.

   (a) In 100 rolls of the dice, what is the probability that five consecutive rolls of seven will occur? Use the five-step method, and model using Monte Carlo simulation.

   (b) What is the average number of rolls until a roll of seven occurs? Use any method.

3. Reconsider the inventory problem of Example 8.1. In the text we stated that if the time between customer arrivals is exponential with mean one, the distribution of the number of arrivals in one week is Poisson with mean one.

(a) Use Monte Carlo simulation to model one week of arrivals. Assume that the time between arrivals is exponential with mean one. Simulate to determine the mean number of arrivals in one week.

(b) Modify the simulation program to keep track of the fraction of the simulated weeks in which there were 0, 1, 2, 3, or more than 3 arrivals. Compare with the Poisson probabilities given in Section 8.1.

4. (a) Repeat Exercise 3, but now assume that the time between customer arrivals is uniformly distributed between 0 and 2 weeks.

(b) Use the probabilities from your computer program output to modify the state transition probabilities for the Markov chain in Example 8.1.

(c) Solve $\pi = \pi P$ for the steady–state probabilities, assuming a uniform customer interarrival time.

(d) Determine the steady–state probability that demand exceeds supply for this modified example.

(e) Compare the results of (d) to the calculations of Section 8.1, and comment on the robustness of our original model with respect to the assumption of random arrivals.

5. Reconsider the docking problem of Example 9.3, but now assume that the time to make a control adjustment is uniformly distributed between 4 and 6 seconds.

(a) Modify the algorithm in Fig. 9.10 to reflect this change in the distribution of c_n.

(b) Implement the algorithm in part (a) on a computer.

(c) Make 20 simulation runs with $k = 0.02$, and tabulate your results. Estimate the mean time to dock.

(d) Compare the results of (c) to those obtained in Section 9.2. Would you say that the model is robust with respect to the assumption that c_n is normally distributed?

6. Reconsider the docking problem of Example 9.3.

(a) Implement the algorithm in Fig. 9.10 on a computer. Make a few model runs, and compare with the results in the text.

(b) Vary the parameter k to determine the optimal value of this control parameter. You will need to make several model runs for each value of k to determine average behavior.

(c) Explore the sensitivity of your answer in part (b) to the initial velocity of the spacecraft, which we assumed to be 50 m/sec.

(d) Explore the sensitivity of your answer in (b) to the docking threshold, which we assumed was 0.1 m/sec.

7. Reconsider the rainy day problem of Example 9.1, but now assume that if today is rainy, there is a 75% chance that tomorrow will be rainy, and likewise, if today is sunny, there is a 75% chance that tomorrow will be sunny. The day we arrived on vacation was sunny ($X_0 = 0$).

   (a) Determine the steady–state probability that any future day will be rainy. Model $\{X_t\}$ as a Markov chain.

   (b) Estimate the probability of three consecutive days of rain using a Monte Carlo simulation.

8. Reconsider the cell division problem of Exercise 11 in Chapter 8.

   (a) Use a Monte Carlo simulation to model the cell division process. How long will it take on average for one cell to grow to 100 cells?

   (b) What is the probability that the family line of one cell will die out before reaching 100 cells?

9. (Continuation of Chapter 7, Exercise 8) Simulate a random arrival process with an arrival rate of five minutes using the Monte Carlo method. Determine the average time between the last arrival prior to $t = 1$ hour and the next arrival after $t = 1$ hour.

10. (Continuation of Chapter 7, Exercise 9) Simulate the supermarket checkout stand problem. Draw a random number to represent your service time, and then see how many random numbers it takes to exceed that. Repeat the simulation a large number of times and determine the average number it takes to find a random number that exceeds yours. Justify the difference between your answer and the one obtained in part (c) of Chapter 7, Exercise 9.

11. Reconsider the inventory problem of Example 8.1, and determine the average number of lost sales per week.

    (a) Assume that on each day a customer arrives with probability 0.2. Construct a Monte Carlo simulation based on a time step of one day to simulate one week of sales activity. For a beginning inventory of 1, 2, or 3 aquariums determine the average number of lost sales by repeated simulation.

    (b) Combine the results of part (a) with the steady–state probabilities calculated in Section 8.1 to determine the overall average number of lost sales per week.

12. This exercise explains the binomial model. Suppose m independent random trials are conducted, each of which has a probability q of success. Let $X_i = 1$ if the ith trial is successful, and $X_i = 0$ otherwise. Then $X = X_1 + \cdots + X_m$ is the number of successes.

(a) Show that $EX = mq$ and $VX = mq(1-q)$. [Hint: First show that $EX_i = q$ and $VX_i = q(1-q)$.]

(b) Explain why there are $\binom{m}{i}$ possible ways for $X = i$ to occur and why each one has probability $q^i(1-q)^{m-i}$.

(c) Explain why Eq. (9.14) in the text represents the probability of i successes in m trials.

13. Reconsider the bombing run problem of Example 9.4. Modify the model to print out the expected number of aircraft lost during this mission. Take into account the possibility that additional planes are lost after the attack is concluded, as the bombers are leaving the target area.

(a) How many planes on average are lost during the mission if $N = 15$ are sent?

(b) Perform a sensitivity analysis with respect to N.

(c) What happens if we can use advanced bombers that fly at 1,200 miles/hour and only need to loiter in the target area for 15 seconds?

(d) Perform a sensitivity analysis on the probability q that one missile kills one plane. Consider $q = 0.4$, 0.5, 0.6, 0.7, and 0.8. State your general conclusions. Under what circumstances would a responsible commander order his pilots to fly this mission?

14. Reconsider the bombing run problem of Example 9.4. Suppose that superior technology allows most bombers to get through the air defense undetected.

(a) Suppose that four aircraft are detected. Let Y denote the number of these aircraft that survive eight shots by the air defense. Determine the probability distribution of Y and calculate the mean number of aircraft lost prior to completion of the attack. Use a Monte Carlo simulation based on a Markov chain model, as discussed at the end of Section 9.3.

(b) Repeat part (a), but now use an analytic simulation.

(c) Suppose that you were required to incorporate the possibility of multiple shots at a single aircraft into the model of Example 9.4. You have two options available to you. You may write a purely analytic simulation incorporating the results of part (b), or you may use a generalized version of the Monte Carlo simulation model of part (a) to obtain the probability distribution of Y for $d = 1, \ldots, 7$ aircraft detected, and incorporate these results into the model as data. Which option would you choose? Explain.

15. (Hard problem) Carry out the model enhancements described in problem 14(c).

16. Reconsider the bombing run problem of Example 9.4.

    (a) Use a Monte Carlo simulation to find the probability of mission success if $N = 15$ aircraft are sent.

    (b) Perform a sensitivity analysis on N. Determine the approximate probability of mission success for $N = 12$, 15, 18, and 21.

    (c) Compare the relative advantages of Monte Carlo and analytic simulations in terms of the difficulty of both model formulation and sensitivity analysis.

17. Reconsider the bombing run problem of Example 9.4. This problem shows how the binomial formula

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i}$$

    can be used to simplify the analytic simulation model presented in the text.

    (a) Use the binomial formula to derive the equation

$$S = 1 - (1-p)^{N-m}(q + (1-p)(1-q))^m.$$

    (b) Show that the number N of planes required to ensure a success probability S is the smallest integer greater than or equal to

$$N = \log\left[\frac{(1-S)(1-p)^m}{(q+(1-p)(1-q))^m}\right] / \log(1-p).$$

    (c) Use this formula to verify the sensitivity analysis results reported in Fig. 9.17.

18. A radio communications channel is active 20% of the time and idle 80% of the time. The average message lasts 20 seconds. A scanning sensor monitors the channel periodically in an attempt to detect the location of emitters using this channel. An analytic simulation model of scanner performance is to be constructed. It would greatly simplify the model if it were at least approximately true that the state of the channel (busy or idle) when one scan is made is independent of the state found during the previous scan. Model the channel using a two–state Markov process and use an analytic simulation to determine how long it takes until the process settles down into steady state. After this point the process essentially forgets its original state.

    (a) Determine the steady–state distribution for this Markov process.

    (b) Derive the set of differential equations satisfied by the state probabilities $P_t(i) = \Pr\{X_t = i\}$. See Section 8.2.

(c) If $X_t = 0$ (channel idle), how long does it take for the state probabilities to get within 5% of their steady–state values?

(d) Repeat part (c), assuming that $X_t = 1$ (channel busy).

(e) How far apart should successive scans be in order for the Markov property to apply (at least approximately) for this model?

19. This problem suggests a way to solve the rainy day problem of Example 9.1 using an analytic model.

(a) Let C_t denote the number of consecutive rainy days by day t. Show that $\{C_t\}$ is a Markov chain. Write down the transition diagram and the transition matrix for this Markov chain.

(b) We are interested in the probability that $\max\{C_1, \ldots, C_7\} \geq 3$. Alter the Markov chain model from part (a) by restricting the state space to just $\{0, 1, 2, 3\}$. Change the state transition probabilities so that 3 is an absorbing state; i.e., set

$$\Pr\{C_{t+1} = 3 | C_t = 3\} = 1.$$

Explain why the probability of at least three consecutive rainy days this week is the same as $\Pr\{C_7 = 3 | C_0 = 0\}$.

(c) Use the methods of Chapter 8 to calculate the probability of at least three consecutive rainy days in one week, assuming a 50% chance of rain each day.

(d) Perform a sensitivity analysis on the 50% assumption. Compare with the results shown in Fig. 9.5.

(e) Compare this analytic model to the Monte Carlo model used in Section 9.1. Which do you prefer, and why? If you had just now come across this problem, which modeling approach would you have selected?

20. Use the method of particle tracking to solve the pollution problem of Example 7.5.

(a) Implement the particle tracking code in Figure 9.23. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.5 of the text.

(b) Modify the code so that the velocity $v = 3.0$ km/hr is a constant. Repeat part (a), and compare to the results of Example 7.5. Are the results consistent with those reported in the text?

(c) Perform a sensitivity analysis on the Monte Carlo simulation results in part (b), to determine how the three measures of performance $Tmax$, $Pmax$ and $Tsafe$ depend on random factors. How confident are you of the results reported in part (b)?

(d) Perform a sensitivity analysis on the plume velocity. Repeat the simulation of part (b) several times for each value of v in Table 7.1, and use the average to reproduce the results in Table 7.1. How confident are you of the tabulated results?

(e) Compare this Monte Carlo simulation model to the analytical model used in Section 7.4. Which do you prefer, and why? If you had just now come across this problem, which modeling approach would you have selected?

21. Reconsider the pollution problem of Example 7.5, but now assume that the wind speed v (km/hr) is given by the formula

$$v = 3 + \frac{M - 3}{1 + 0.1d^2} \tag{9.29}$$

where d (km) is the distance from the center of town, and $M = 8$ km/hr is the wind speed in the center of town.

(a) Plot the new wind speed function together with the wind speed function from Example 7.5. Do they seem comparable?

(b) Implement the particle tracking code in Figure 9.23. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.5 of the text.

(c) Modify the code in part (b) using the new wind speed. Repeat part (b) and compare the results. Is the model of Example 7.5 robust with respect to the assumed wind speed function?

(d) Perform a sensitivity analysis on the maximum wind speed in town. Repeat part (c) for $M = 4, 6, 8, 10, 12$ km/hr. How sensitive are the three measures of performance, $Tmax$, $Pmax$ and $Tsafe$, to the maximum wind speed?

22. This problem investigates a power law model for dispersivity in the pollution problem of Example 7.5.

(a) Implement the particle tracking code in Figure 9.23. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.5 of the text.

(b) Modify the code in part (a) to compute the sample variance $s^2(j)$ of the particle location $\{S(i, j) : 1 \leq i \leq N\}$ at each time $t(j)$, for $j = 1, 2, \ldots, M$.

(c) Plot the variance $s^2(j)$ versus time $t(j)$ from part (b), similar to Figure 9.27. Comment on main features of the graph.

(d) Make a log-log plot of the variance $s^2(j)$ versus time $t(j)$ from part (b). That is, plot $\log s^2(j)$ versus $\log t(j)$. Do the points on this log-log plot seem to follow a straight line?

(e) Fit a power law model $\sigma = Ct^p$ to the results of part (b). One way to do this is to apply linear regression to the log-transformed data, compare Exercise 18 in Chapter 8. Plot the power law model from part (e) together with the variance data from part (c). Is it reasonable to adopt a power law model for the variance of the particle plume? What is the practical utility of this model?

23. This problem introduces the gamma function, and the Laplace transform, in the context of fractional derivatives. The gamma function is defined by

$$\Gamma(b) = \int_0^\infty x^{b-1} e^{-x} dx$$

for $b > 0$. The Laplace transform of a function $f(x)$ is defined by

$$F(s) = \int_0^\infty e^{-sx} f(x)\, dx.$$

(a) Use integration by parts to show that $\Gamma(b+1) = b\Gamma(b)$. Conclude that $\Gamma(n+1) = n!$

(b) Use a substitution $y = sx$ to show that the function $f(x) = x^p$ has Laplace transform $s^{-p-1}\Gamma(p+1)$.

(c) Use integration by parts to show that $sF(s) - f(0)$ is the Laplace transform of the first derivative $f'(x)$.

(d) The Caputo fractional derivative of order $0 < \alpha < 1$ has Laplace transform $s^\alpha F(s) - s^{\alpha-1} f(0)$. Use this formula to show that $f(x) = x^p$ has fractional derivative

$$\frac{\Gamma(p+1)}{\Gamma(p+1-\alpha)} x^{p-\alpha}.$$

(e) Explain why the same formula also holds for positive integers α.

24. This problem introduces two integral formulas for fractional derivatives. The Riemann-Liouville fractional derivative of order $n-1 < \alpha < n$ is defined by

$$\frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_0^\infty f(x-y) y^{n-\alpha-1} dy, \tag{9.30}$$

using the gamma function introduced in Exercise 23. The Caputo fractional derivative of order $n-1 < \alpha < n$ is defined by

$$\frac{1}{\Gamma(n-\alpha)} \int_0^\infty \frac{d^n}{dx^n} f(x-y) y^{n-\alpha-1} dy, \tag{9.31}$$

by moving the derivative inside the integral.

(a) Apply the formula (9.30) to compute the Riemann-Liouville fractional derivative of the function $f(x) = e^{ax}$ for $a > 0$.

(b) Explain why the resulting formula is also true for positive integers α.

(c) Apply the formula (9.31) to compute the Caputo fractional derivative of the function $f(x) = e^{ax}$ for $a > 0$. Compare with the answer in part (a).

(d) Apply the formula (9.30) to compute the Riemann-Liouville fractional derivative of the constant function $f(x) = 1$ for all $x \geq 0$.

(e) Apply the formula (9.31) to compute the Caputo fractional derivative of the constant function $f(x) = 1$ for all $x \geq 0$. Compare with the answer in part (d). Which formula agrees with the case where α is a positive integer?

25. Reconsider the water pollution problem of Example 9.6, and now consider a simple analytical model for downstream contamination at early time.

(a) Implement the particle tracking code in Figure 9.31. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.6 of the text.

(b) Modify the code to estimate the *earliest* time $T\,risk$ that downstream concentration at the point $x = 20$ m downstream exceeds 2 Ci.

(c) Repeat part (b) several times to get an average value for $T\,risk$. How accurate is this value?

(d) Repeat part (c) for $x = 25, 30, 35, 40$ m downstream. Plot $T\,risk$ versus x.

(e) It was stated in the text that the α-stable density $C(x,t) \approx t\alpha A x^{-\alpha-1}$ for x sufficiently large. Use this asymptotic approximation to estimate $T\,risk$. Compare to the results of the Monte Carlo simulation. Does the analytical formula provide a reasonable estimate?

26. Reconsider the water pollution problem of Example 9.6, and perform a sensitivity analysis on the tail parameter α.

(a) Implement the particle tracking code in Figure 9.31. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.6 of the text.

(b) Repeat part (a) with $\alpha = 1.2, 1.3, 1.5, 1.8$. How sensitive are the results to the tail parameter α?

(c) Repeat part (b) several times for each value of $\alpha = 1.1, 1.2, 1.3, 1.5, 1.8$. Tabulate the average value of each measure of performance, $T\,max$, $P\,max$ and $Tsafe$.

(d) Give an accuracy estimate for the numbers you tabulated in part (c).

(e) Estimate the sensitivities $S(T\,max, \alpha)$, $S(P\,max, \alpha)$ and $S(Tsafe, \alpha)$, and interpret in the context of this problem.

27. Reconsider the water pollution problem of Example 9.6. Some scientists believe that there is a natural upper bound to the magnitude of particle jumps.

    (a) Implement the particle tracking code in Figure 9.31. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.6 of the text.

    (b) Modify the code to enforce a maximum particle jump which is $J = 100$ times the average jump size. Repeat part (a) and compare the results.

    (c) Repeat part (b) several times, and tabulate the average value of each measure of performance, $Tmax$, $Pmax$ and $Tsafe$.

    (d) Repeat part (c) with $J = 25$. How does the maximum jump size affect the results?

28. Reconsider the water pollution problem of Example 9.6, and consider how the tail parameter α affects the distribution of particle location.

    (a) Implement the particle tracking code in Figure 9.31. Run the code, and verify that the outputs are reasonably consistent with those reported in Example 9.6 of the text.

    (b) Plot a relative frequency histogram of particle locations at time $t = 224$ days, similar to Figure 9.32. Repeat several times, and comment on how the histogram shape varies depending on random factors.

    (c) Repeat part (b) for $\alpha = 1.2, 1.3, 1.5, 1.8$. How does the distribution of particle locations vary with the tail parameter α?

    (d) Repeat part (b) with $\alpha = 4$. How does the resulting histogram shape compare?

    (e) What does the central limit theorem of Section 7.3 imply about the distribution of particle location in part (d)?

29. This problem explores the fractal particle traces discussed in Section 9.5.

    (a) Implement the particle tracking code in Figure 9.31 with $N = 20$ to obtain simulated particle locations $S(i, j)$ at times $t(j)$ for particles $i = 1, 2, \ldots, N$.

    (b) Use the results of part (a) to plot the particle traces for $N = 20$ particles, similar to Figure 9.26. Comment on the features of these plots. How do they compare to the particle traces in Figure 9.26?

    (c) Repeat part (b) for $\alpha = 1.2, 1.3, 1.5, 1.8$, and recall from Section 9.5 that each graph is a fractal of dimension $2 - 1/\alpha$. How does the graph vary with the parameter α?

    (d) Modify the code to produce two sets of traces for each particle, representing the x and y coordinates of particle location. Plot the y location versus the x location, to get a graph similar to Figure 9.35.

(e) Repeat part (d) for $\alpha = 1.2, 1.3, 1.5, 1.8$, and recall from Section 9.5 that each graph is a fractal of dimension α. How does the appearance of the graph change as α increases?

Further Reading

1. Benson, D. A., Schumer, R., Meerschaert, M. M. and Wheatcraft, S. W. (2001) Fractional dispersion, Lévy motion, and the MADE tracer tests, *Transport in Porous Media* Vol. 42, 211–240.

2. Boggs, J. M., Beard, L. M., Long, S. E. and McGee, M. P. (1993) Database for the second macrodispersion experiment (MADE-2), EPRI report TR-102072, Electric Power Res. Inst., Palo Alto, CA.

3. Bratley, P. et al. (1983) *A Guide to Simulation.* Springer-Verlag, New York.

4. Friedman, A. (1975) *Stochastic differential equations and applications. Vol. 1*, Academic Press, New York.

5. Hoffman, D. *Monte Carlo: The Use of Random Digits to Simulate Experiments.* UMAP module 269.

6. McCulloch, J. H. (1997) Measuring tail thickness to estimate the stable index α: a critique. *J.* Business Econom. Statist. Vol. 15, 74-81.

7. Meerschaert, M. and Cherry, W. P. (1988) Modeling the behavior of a scanning radio communications sensor. *Naval Research Logistics Quarterly*, Vol. 35, 307–315.

8. Meerschaert, M. M. and Sikorskii, A. (2012) *Stochastic Models for Fractional Calculus.* De Gruyter Studies in Mathematics **43**, De Gruyter, Berlin.

9. Molloy, M. (1989) *Fundamentals of Performance Modeling.* Macmillan, New York.

10. Press, W. et al. (1987) *Numerical Recipies.* Cambridge University Press, New York.

11. Ross, S. (1985) *Introduction to Probability Models.* 3rd ed., Academic Press, New York.

12. Rubenstein, R. (1981) *Simulation and the Monte Carlo Method.* Wiley, New York.

13. Shephard, R., Hartley, D., Haysman, P., Thorpe, L. and M. Bathe. (1988) *Applied Operations research: Examples from Defense Assessment.* Plenum Press, London.

14. Wheatcraft, S. W. and Tyler, S. (1988) An explanation of scale-dependent dispersivity in heterogeneous aquifers using concepts of fractal geometry, *Water Resources Research*, Vol. 24, 566–578.